Modelling of the Nonstationary Electromyogram

by

Lam, Hok-Hoi

A thesis
presented to the University of Manitoba
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering
in
Department of Electrical Engineering

Winnipeg, Manitoba

ISBN 0-315-37301-6

MODELLING OF THE NONSTATIONARY ELECTROMYOGRAM

BY

HOK-HOI LAM

A thesis submitted to the Faculty of Graduate Studies of
the University of Manitoba in partial fulfillment of the requirements
of the degree of

MASTER OF SCIENCE

This thesis is dedicated to my beloved grandpa.

ACKNOWLEDGEMENTS

I wish to express my special thanks to my supervisor, Dr. E. Shwedyk, for his patient supervision, guidance and fruitful suggestions throughout this research.

It also behooves me to thank for the assistance I have received from my colleagues and friends, especially Mr. Eric Brusse.

Finally, in all sincerity, my parents deserve much more than a simple thanks for their understanding, support, and encouragement over the years.

ABSTRACT

This thesis examines the nonstationary electromyogram(EMG) by describing the experimentally obtained EMG variance and autocorrelation with mathematical functions.

Based on Kreifeldt's postulation, which considers the EMG signal to be an amplitude modulated signal where the carrier is a random process and the number of active motor units is the modulating signal, two mathematical functions are used to curve-fit the EMG variance and one to curve-fit the autocorrelation. Performance of these functions are evaluated using the mean-square-error criterion. Results have shown that these functions describe the EMG variance and autocorrelation well. The two variance functions used to curve-fit the EMG variance have errors which ranged from 0.67% to 7.87% while the function used to curve-fit the autocorrelation has errors which ranged from 4.23% to 28.41%. Finally, the Midpoint Moving Average Estimator and the Homomorphic filter are developed to estimate the EMG variance.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter I

INTRODUCTION

The purpose of this thesis is to experimentally examine the nonstationary electromyogram (EMG), in order to gain more insight regarding its use as a control signal in devices such as prosthetic arms.

Over the past two decades, design of controllers for electrically powered prosthetic devices has treated the EMG as a stationary signal. The most sophisticated prosthetic device has a multi-channel or multi-state controller, and has the capability of producing six motions[Saridis,1982]. This prosthesis is slow in responding to EMG signal since the EMG must become stationary at a specified level before a motion is produced. Therefore, the control of prosthetic devices may be improved if nonstationary EMG can be utilized as a control signal.

It has been postulated that the EMG signal e(t) can be considered as an amplitude modulated signal[Kreifeldt,1974]. The main purpose of this thesis is to verify further by experimental work this amplitude modulated model of EMG signal generation. Three mathematical models are proposed and investigated to describe the variance and autocorrelation of EMG. Experimental data were obtained to evaluate the per-

formance of these functions using the mean-square-error criterion. These functions are used to develop two variance estimators, the Midpoint Moving Average Estimator(MMAE)[Xiong,1985] and a homomorphic filter.

The thesis consists of five chapters. Chapter two reviews the physiological structure of a muscle, and the generation and properties of the EMG. Chapter three describes the experimental procedure for acquiring data and subsequent data processing. Chapter four discusses results as well as application of the functions to the MMAE and homomorphic filters. Finally, conclusions and recommendations are given in Chapter five.

Chapter II

BACKGROUND

This chapter briefly reviews the necessary background material. Section 2.1 reviews the physiological structure of muscle and how the EMG is generated. Section 2.2 describes the relevant EMG properties and introduces a mathematical model for the EMG.

2.1 PHYSIOLOGICAL BACKGROUND

The following discussion is based on the physiology text of Crouch[1972]. Muscle contraction, controlled by the nervous system, generates human movements and also the EMG signal. There are two types of contractions, isotonic and isometric. Isotonic contraction produces movements and involves the shortening and lengthening of muscle fibres. Isometric contraction does not produce any movement but provides fixed gestures, i.e., the muscle length remains constant. Due to the fact that the properties of the EMG signal partly depend on the length, velocity and shortening of muscle, and that the EMG is typically generated by an isometric contraction for control purpose, isometric contraction was chosen to generate the EMG signals in this thesis.

All muscles are composed of elongated cells called muscle fibers. These muscle fibres contain fine fibrils called myofibrils within their cytoplasm, also called sarcoplasm. Three types of muscle tissues are found in the human body, smooth, cardiac and skeletal. Skeletal muscles are also called striated muscles due to their longitudinally arranged myofibrils. More importantly, they are the muscles that are voluntarily controllable; therefore, typically, they are chosen for EMG study.

Each nerve fibre innervates from a few to hundreds of skeletal muscle fibres. The nerve fibres along with the innervated muscle fibres constitute a motor unit. When a motor unit is stimulated by nerve impulses, the corresponding muscle fibres contract and generate a force.

An action potential, physiologically called a nerve impulse, may be defined as a physiochemical change in nerve fibres which once initiated, is self-propagating. It can last for a period of 5 ms, and can travel along the cell membrance at velocities of up to 120 meters per second. Therefore, it is possible to have a sequence of action potentials travelling along a nerve fibre. When this sequence of action potentials reaches its corresponding muscle fibres, it causes a contraction in the muscle fibres. The transmission of these action potentials along the muscle fibres produces an electrical signal commonly known as the EMG signal which can be detected by surface electrodes. Its

characteristics depend on the number of motor units being stimulated in a muscle and the frequency of the action potential train. Previous studies have shown that this EMG signal is a zero-mean Gaussian signal, even when it is generated by a low level muscle contraction[Shwedyk, 1974].

2.2 THEORETICAL BACKGROUND

As has been mentioned, the EMG can be modelled as an amplitude modulated signal. Thus, the EMG signal e(t) can be expressed as

$$e(t)=n(t)w(t). \qquad (2.1)$$

The modulating signal n(t) represents the number of active motor units. The carrier w(t) is a random process which is assumed to be stationary. Because the EMG is a zero-mean Gaussian random process, w(t) can be assumed to be zero-mean Gaussian with unit variance. In the case of stationary EMG, n(t) is just a constant, while in the nonstationary case, it varies as a deterministic function of time.

The autocorrelation function reveals the dependence of a signal at two time instances. For the EMG e(t), it is given by:

$$\begin{aligned} R_{ee}(t+\tau,t) &= E[e(t+\tau)e(t)] \\ &= E[n(t+\tau)n(t)]\ E[w(t+\tau)w(t)] \\ &= n(t+\tau)\ n(t)\ R_{ww}(\tau). \end{aligned} \qquad (2.2)$$

where E[ ] is the expectation operator. For stationary EMG, the above expression becomes

$$R_{ee}(\tau)=K\ R_{ww}(\tau), \tag{2.3}$$

where K is a constant. Parker [Parker, 1977] has found that the autocorrelation function of w(t), $R_{ww}(\tau)$, can be described by the following function:

$$R_{ww}(\tau)=\left(\frac{1}{\alpha^3}+\frac{|\tau|}{\alpha^2}-\frac{\tau^2}{\alpha}\right)\ \exp[-\alpha|\tau|] \tag{2.4}$$

where α is a constant depending on the physiology of the muscle. The curve of this function has one main-lobe and two small-side lobes. It is worthwhile to mention that this expression was derived under the following assumptions:

1. All motor units are uncorrelated.
2. All motor unit action-potential waveforms are identical.
3. All muscle-fibre propagation velocities are identical.

In the case of nonstationary EMG, when $\tau=0$, function 2.2 becomes

$$R_{ee}(t,t)=n^2(t)\ E[w^2(t)], \tag{2.5}$$

Since the EMG has zero mean and w(t) has been assumed to have unit variance,

$$R_{ee}(t,t)=n^2(t).$$

Thus, the EMG variance is,

$$\sigma^2_e(t)=n^2(t) \qquad (2.6)$$

Two heuristic functions were used to curve-fit experimentally obtained variance data. They are:

1. $$\sigma^2_{e_1}(t)=K[1-\exp(-K_1 t)], \qquad (2.7)$$

2. $$\sigma^2_{e_2}(t)=K[1-A_1\exp(-At)-B_1\exp(-Bt)] \qquad (2.8)$$

The constant K is simply a scaling factor. Parameters K_1, A and B are estimated according to certain criteria given later in this chapter. Since the first derivative of function 2.8 is constrained to zero at $t=0$, A_1 and B_1 are variables depending on parameters A and B. These two functions are chosen to reflect the fact that the EMG variance always rises smoothly from one level to another, i.e., it would not jump from one level to another level as a step function would.

Equation 2.7 is a simpler expression. One disadvantage of this function is that it tends to have more error at the lower level of the EMG variance as shown in Figure 2.1. However, it does perform well at the higher levels. In order to minimize the lower level error while still keeping the good performance at high level, function 2.8 was intro-

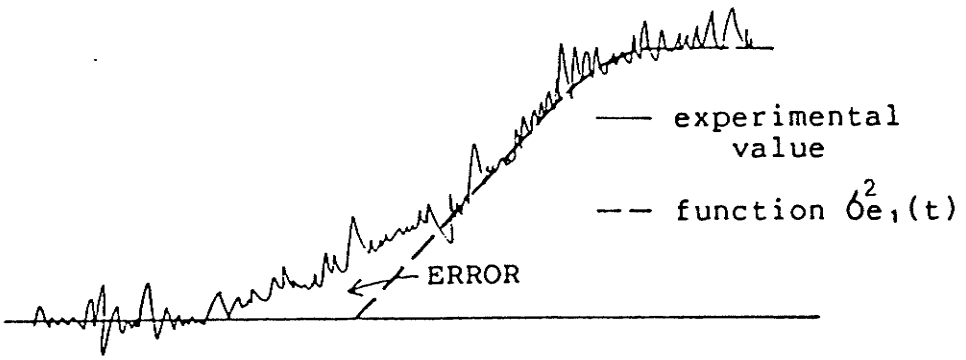


Figure 2.1: Low Level Error of Function $\delta^2_{e_1}(t)$

duced. The mean-square-error criterion is used here to evaluate the performance of the variance functions in curve-fitting the experimentally obtained results.

Chapter III

EXPERIMENTAL WORK

This chapter describes the experimental study used for data generation and acquisition and the signal analysis performed. The entire experimental set-up is first reviewed in section 3.1. The appropriate terminology and how the data-acquisition was performed is described next in section 3.2. Section 3.3 explains the data-processing scheme to compute the EMG variance and autocorrelation of the experimentally obtained EMG data, and also explains programs that curve-fit the functions to the experimental EMG variance and autocorrelation.

3.1 EXPERIMENTAL SET-UP

The entire experimental set-up for data-acquisition may be best illustrated in terms of the block diagram in figure 3.1.

Two channels of data are sampled simultaneously by the PDP-11/40 system. One channel is the EMG signal, while the other is the strain-gauge signal which represents the force produced by the muscle. Each channel is sampled at a sampling frequency of 500Hz.

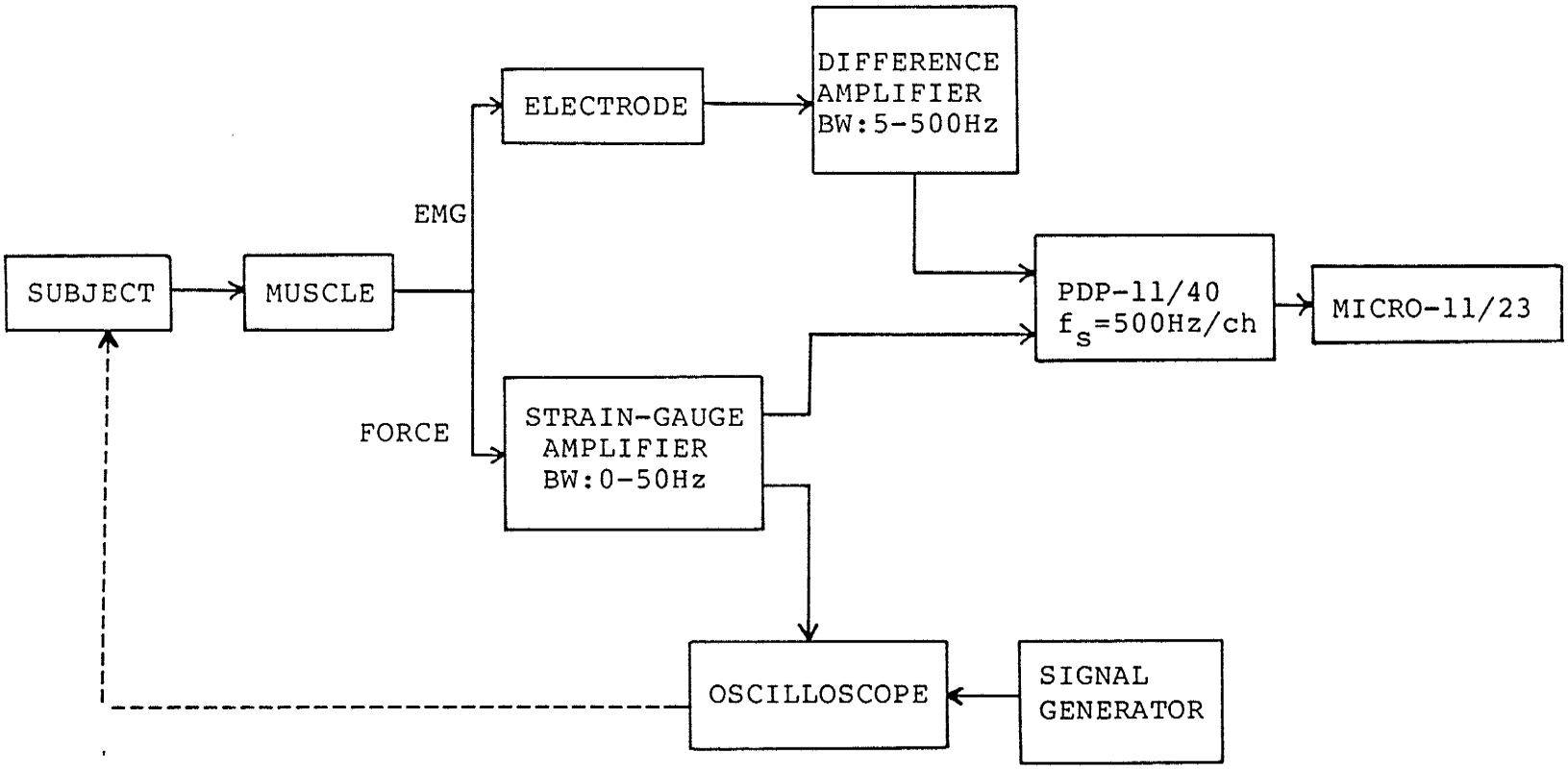


Figure 3.1: Experimental Set-up for Data Acquisition

This sampling frequency is determined by the EMG power density spectrum which lies between 10Hz and 200Hz. The highest frequency of the strain-gauge signal is less than 20Hz.

By means of an oscilloscope, a signal generator is used to indicate to the subject when to contract and when to relax. The oscilloscope displays a continuous square pulse signal at 0.325Hz, that is 6.5 pulses per 20 seconds, with a 40% duty cycle. Other frequencies and duty cycles ranging from 3 to 10 pulses per 20 seconds with duty of 30% to 60% have also been tested. The consensus of the subjects was that the chosen frequency and duty cycle was the most comfortable. The pulse magnitude used to specify the subject's contraction level was set to 35% of the subject's maximum strength. It was found that, for a level of 70% or more of the subject's maximum strength, the muscle fatiqued rapidly; and for a 50% level, the subject had difficulty maintaining a constant contraction for the required time duration. Therefore, 35% was chosen. This level produced a large enough EMG for reliable data acquisition, yet did not fatique the muscle.

A strain-gauge device was designed to measure the force produced by the muscle. This device consists of strain-gauges arranged in a Wheatstone bridge, a low-pass filter and a low noise amplifier. Because the shortest time needed to contract a muscle from one force level to another is usually longer than 100ms, which implies a signal bandwidth of

less than 10Hz, the low-pass filter was chosen to have a cut-off frequency of 50Hz. The amplifier output was connected to the PDP-11/40 for data-sampling and to the oscilloscope as feed-back for the subject.

On the oscilloscope, with proper triggering, the subject saw only two lines, line A and line B. Assume that line A and line B are controlled, respectively, by the signal generator and the strain-gauge amplifier. These two lines go either high or low, since both the strain-gauge and the square pulse signals are very low frequency signals. To produce a muscle contraction, when the subject saw line A go high, he contracted his muscle to bring line B to match with line A as quickly as possible; and when line A went low, he relaxed and waited for the next trial. For the experiment, the subject was instructed to control line B so that it would not fluctuate about line A to any large degree. If the fluctuation range was greater than 10% of the specified contraction level, the data-file was simply discarded. The purpose of doing these was to ensure that each trial of the experiment was as repeatable and consistent as possible.

Surface electrodes made of silver were used to detect the EMG. Along with a Ag-Cl base jell, the electrode system proved reliable and capable of eliminating motion artifact. They were placed on the subject thirty minutes before the experiment started to allow the impedance of the interface between the electrodes and skin to stablize. In order to

decrease the effect of 60Hz interference from the power line, electrodes were connected to the difference amplifier via coaxial cable. Occasionally, if the 60Hz interference was too excessive, the subject was asked to hold an additional ground wire.

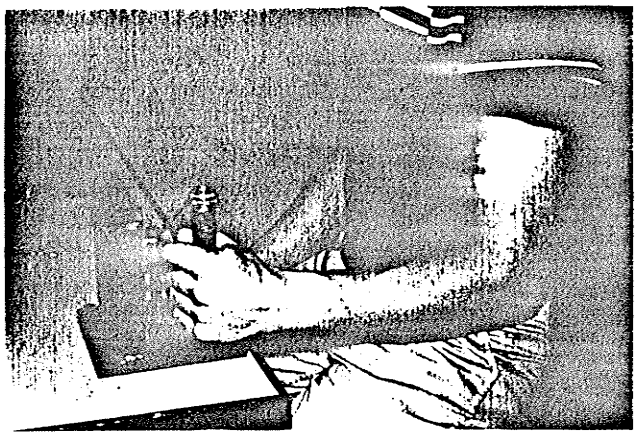
The instrumentation amplifier is a University of Manitoba design. It consists of a frequency adjustable band-pass filter and an high gain amplifier with variable gain control. The band-pass filter was adjusted to pass signals in 5Hz to 500Hz range. The reason for this is again due to the nature of the EMG power density spectrum. Further, since the A/D convertor of the PDP-11/40 digitizes signals within a ±1V range, the amplifier gain was set to 20,000, sometimes 50,000 for some exceptionally small EMG signals.

3.2 DATA ACQUISITION

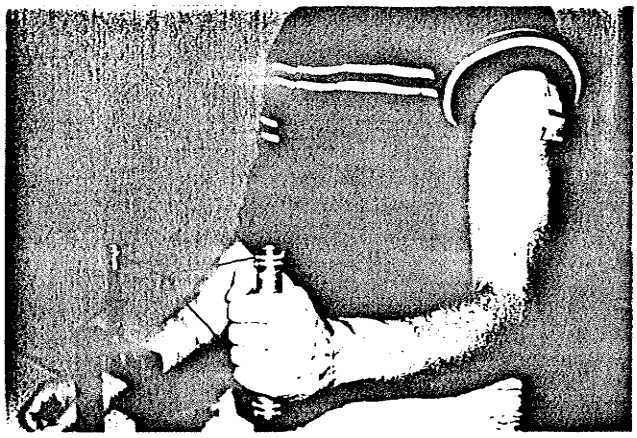
Subjects were selected from the typical university population with ages ranging from 20-40 years. Three different muscles were chosen for the experiment. They were the biceps brachii, the deltoid, and the rectus femoris. These muscles are subcutaneous muscles whose EMG can be easily detected by the surface electrodes. Since the biceps brachii is relatively the easiest to control, it played a major role in this thesis.

Three motions, as shown in figure 3.2, were selected to generate the EMG data. These motions were supination for the biceps brachii, arm-adduction for the deltoid and knee extension for the rectus femoris, see figure 3.2.

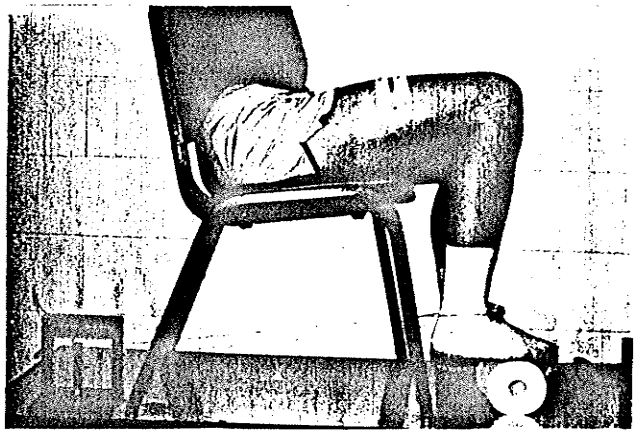
For supination, the subject had to keep his arm vertical to the ground and perpendicular to his fore-arm. When performing the experiment, the subject rotated a metal bar connected to the strain-gauge. For arm-adduction, again, the subject had to keep his arm vertical to the ground and not to rotate his arm when doing the experiment. A wire was attached as close as possible to the subject's elbow joint from the strain-gauge. For knee extension, the subject sat on a chair with his foot on a cylinder. This cylinder made the knee extension easier to control. The wire from the strain-gauge was attached to the subject's foot, slightly above the ankle joint.



- Supination with biceps brachii



- Arm adduction with deltoid



- Knee extension with rectus femoris

Figure 3.2: Three Motions Used in Acquiring Data

To begin data-acquisition, the subject was given a verbal start command. He then tracked the square-pulse signal on the oscilloscope by contracting his muscle in order to match the strain-gauge signal with the square-pulse signal. The subject had to maintain the contraction for about one second, each contraction being a trial. Because of hardware limitations, the computer could only sample six consecutive trials of data for a duration of 20 seconds each time. After six trials which made up a data file, the subject would stop and relax for 30 seconds after which time the whole procedure would be repeated.

Statistically speaking, a complete set of data from one subject is called an ensemble. There are two different ensembles in this thesis. One contains 120 trials called a small-ensemble while the other one contains 600 trials called a big-ensemble. Statistically, the 120 trial-ensemble gives reliable enough information on how the EMG behaves; the 600 trial-ensemble is used to verify the analysis done with the 120 trial-ensemble. As will be seen in subsequent chapters, the variance and autocorrelation curves obtained from the 120 trial-ensembles behave in the same way as those obtained from the 600 trial-ensembles, the only difference being that the 600 trial-ensembles give smoother curves. Eight small-ensembles and three big-ensembles of data were obtained from the biceps brachii, while two small-ensembles of data were obtained from the deltoid and two

small-ensembles from the rectus femoris for comparison purposes.

During the data acquisition of the big-ensemble, if the subject felt that his muscle was exhausted, he could relax for two to three minutes to allow his muscle to recover. Quite often, the subject asked for this recovery period after 150-200 contractions.

After the experiment, the data was transmitted to another computer, MICRO-11/23, for processing. Here, data was first converted to integer form ranging from 0 to 4096 in value. After this conversion, all data was screened on a monitor to ensure that each file contained six trials of data and that no errors occured during data transmission. The data was then ready for further processing.

3.3 DATA-PROCESSING

Computation related to the experimental data was divided into two stages. The first stage, done in the MICRO-11/23 computer, calculated the variance and autocorrelation of the experimental EMG data. The curve-fit of experimental variance and autocorrelation data by functions shown in equations 2.4, 2.7 and 2.8 was then accomplished on the Amdahl mainframe computer.

3.3.1 Computation in MICRO-11/23

After the data was converted to integer, it was submitted to the processing steps outlined in figure 3.3. Each box represents a program. These programs are listed in Appendix B.1.

Program SORT sorts each data file into two different data files; one containing strain-gauge data, the other EMG data. For the strain-gauge data files, the average mid-point of an ensemble is first calculated in program MIDPOINT by finding the average highest and the average lowest points of each trial using the following expression:

$$\overline{\text{mid-point}} = (\overline{\text{highest}} + \overline{\text{lowest}})/2 \qquad (3.2)$$

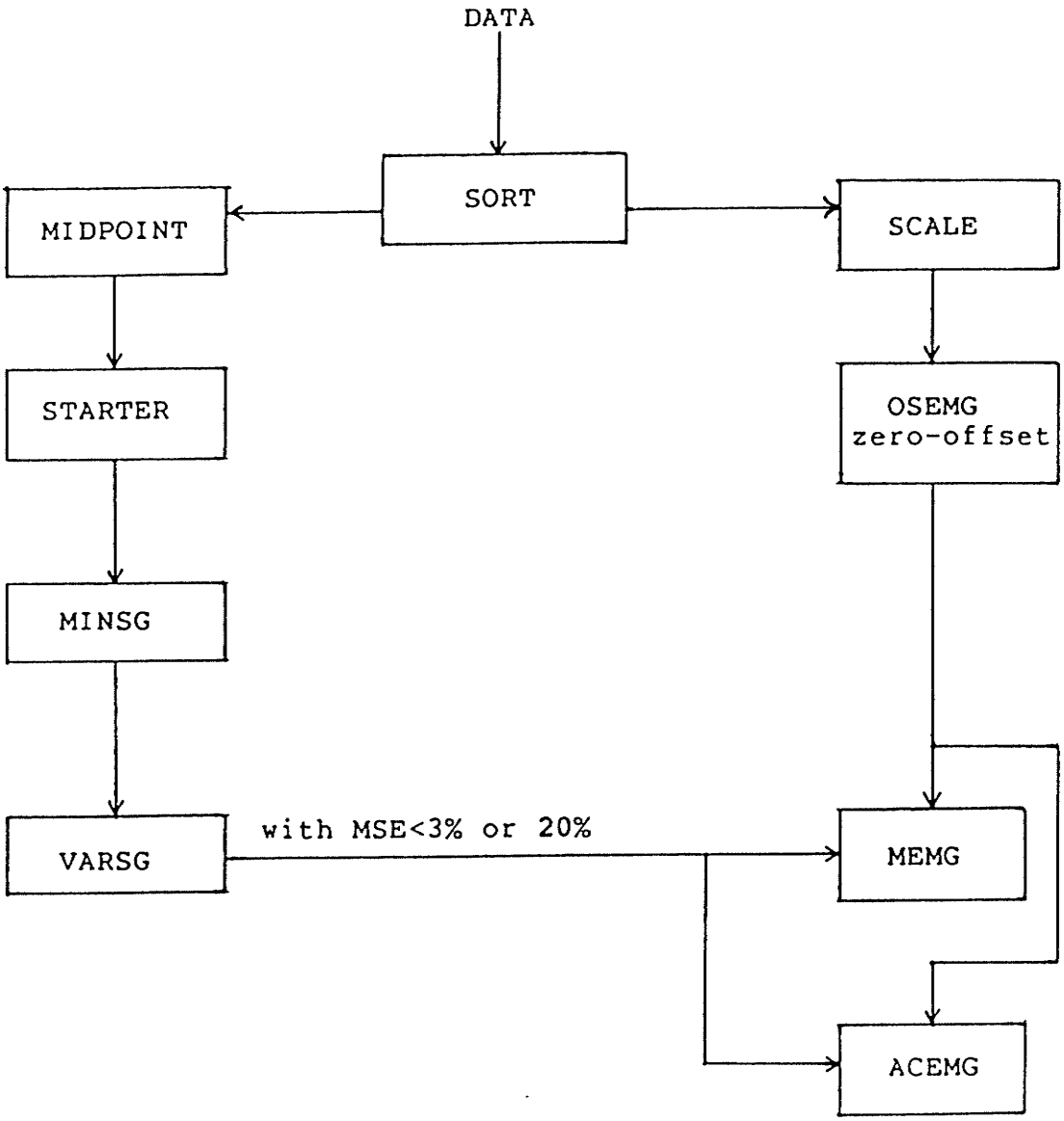


Figure 3.3: Processing Scheme in MICRO-11/23

The starting-point of each trial is located by detecting the mid-point of the trial and going back d samples (see figure 3.4). This is done in program STARTER. Obviously, this starting-point is not the exact starting point where the subject begins to contract the muscle; it represents a standardized reference point for computation. The d samples are also chosen to ensure that this starting-point is before the exact starting point.

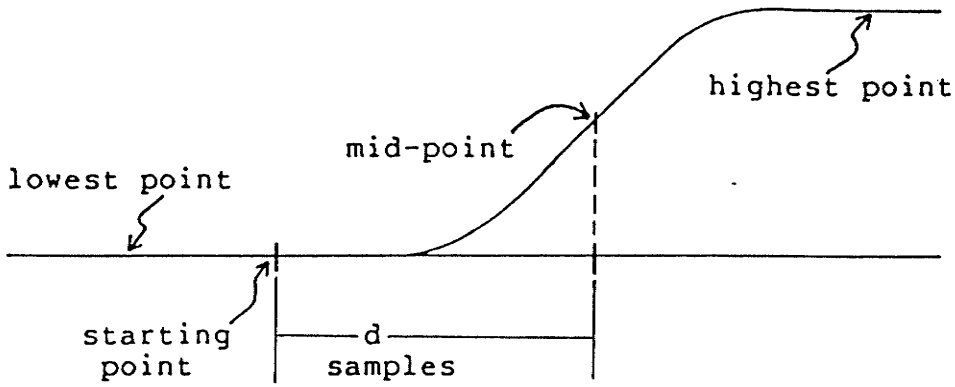


Figure 3.4: Midpoint & Starting Point of a Strain-gauge Signal

After the starting-point of each trial has been located, the average strain-gauge signal of the entire ensemble is computed by the program MINSG. This program utilizes the following expression to calculate the ensemble mean strain-gauge output:

$$\overline{Sg}(n) = \frac{1}{N} \sum_{i=1}^{N} Sg_i(n), \qquad n=0,1,2,... \qquad (3.3)$$

where i is the trial number, N is the total number of trials and n is the sample number. A typical calculated ensemble mean strain-gauge curve is shown in the figure 3.5(also see Appendix C.1).

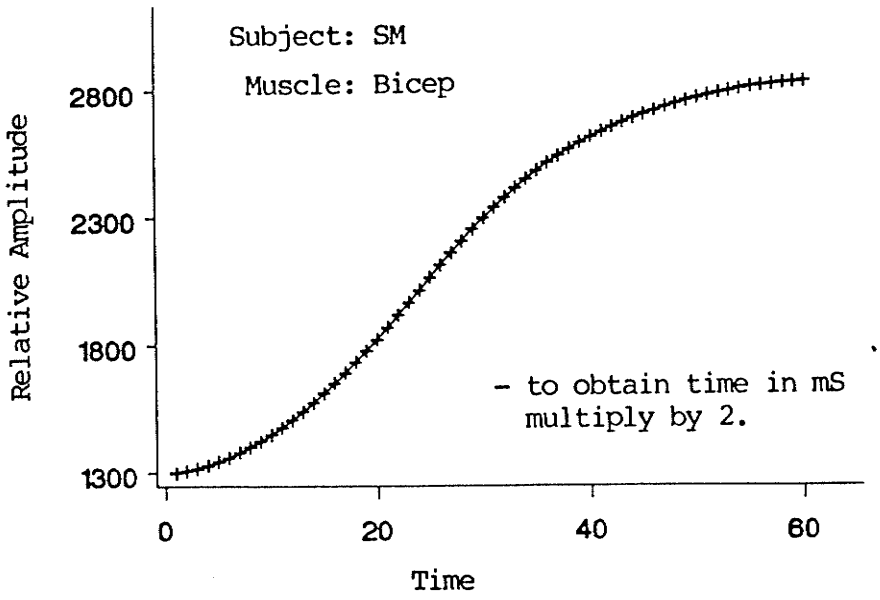


Figure 3.5: A Typical Ensemble-Mean Strain-Gauge Signal

In order to ensure that data chosen for later computation are consistent, the program VARSG computes the mean-square-error of each trial by the following expression:

$$MSE=\left[\sum_{j=1}^{m} \frac{[\overline{Sg}(j)-Sg(j)]^2}{[\overline{Sg}(j)]^2}\right] X\ 100\% \qquad (3.4)$$

This criterion is used to determine which trial of data is included in the computation of EMG variance and autocorrelation. If any trial has an error of 3% error or less when compared with the ensemble mean of the strain-gauge, its corresponding EMG data is selected.

The EMG data is first converted in program SCALE to a real number within the range of ±1V and is then subtracted by the overall mean value in program OSEMG.

The nonstationary EMG variance is then computed in the program MEMG by averaging across the ensemble,

$$\bar{e}^2(t) = \frac{1}{N} \sum_{i=1}^{N} e_i^2(t). \qquad (3.5)$$

This is an unbiased variance estimate, where i is the trial number, N is the total number of trials and t is the time instant at which the variance is estimated. Similarly, the program ACEMG computes the autocorrelation of the EMG using the following expression.

$$R_{ee}(t,\tau) = \frac{1}{N} \sum_{i=1}^{N} e_i(t)\ e_i(t+\tau) \qquad (3.6)$$

where t is a reference point of time, τ varies from +20 to -20 msec and N is the total number of trials. The range of ±20 msec is chosen to reveal both the main and side lobes of the autocorrelation curve. Twenty to thirty different values of t's are used to compute the autocorrelation curves for each ensemble. Some of the computed ensemble mean variance and autocorrelation curves are shown in figure 3.6 and 3.7. Appendix C.2 contains the complete set of curves.

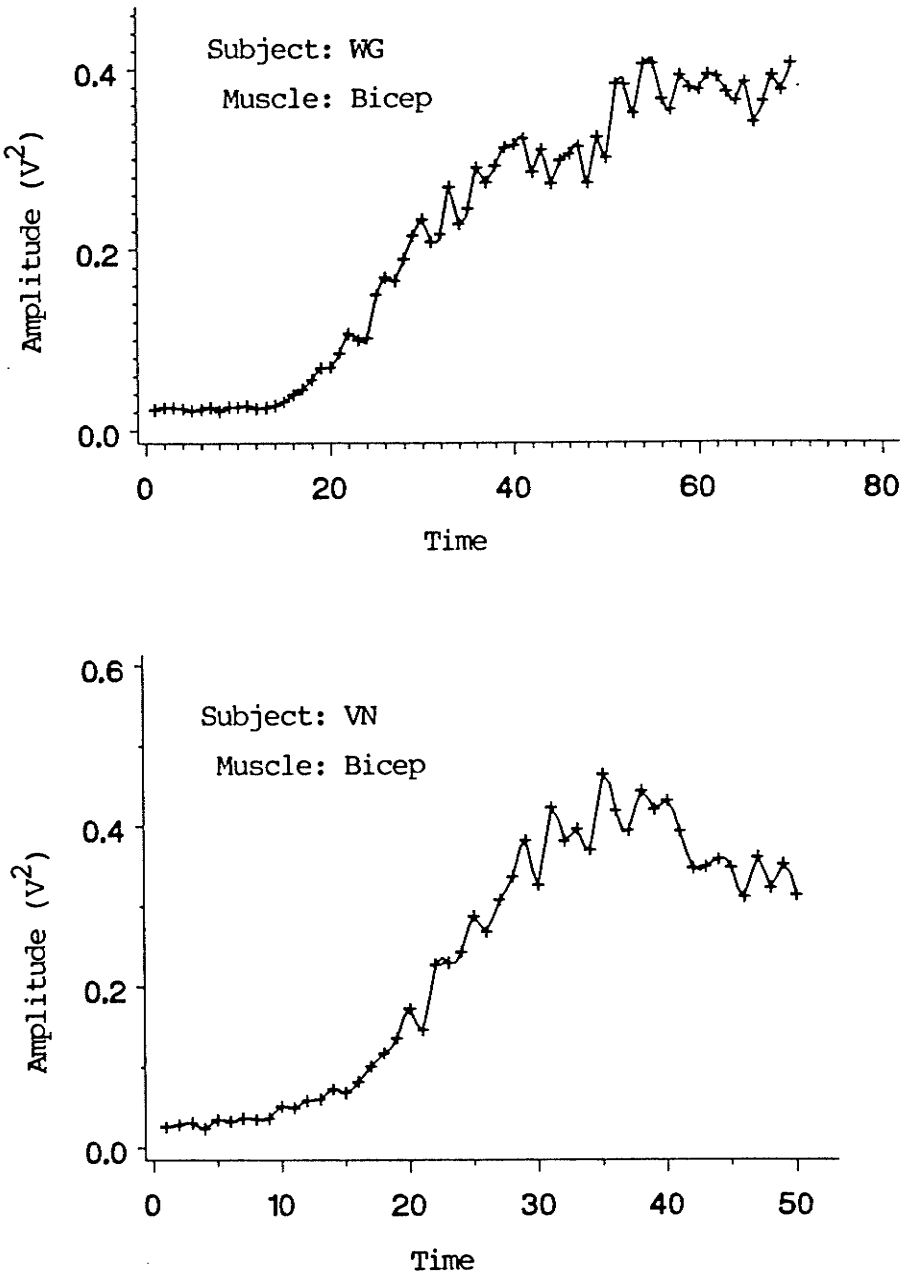


Figure 3.6: Ensemble-Mean Variance of Experimental EMG
- to obtain time in mS multiply by 2.

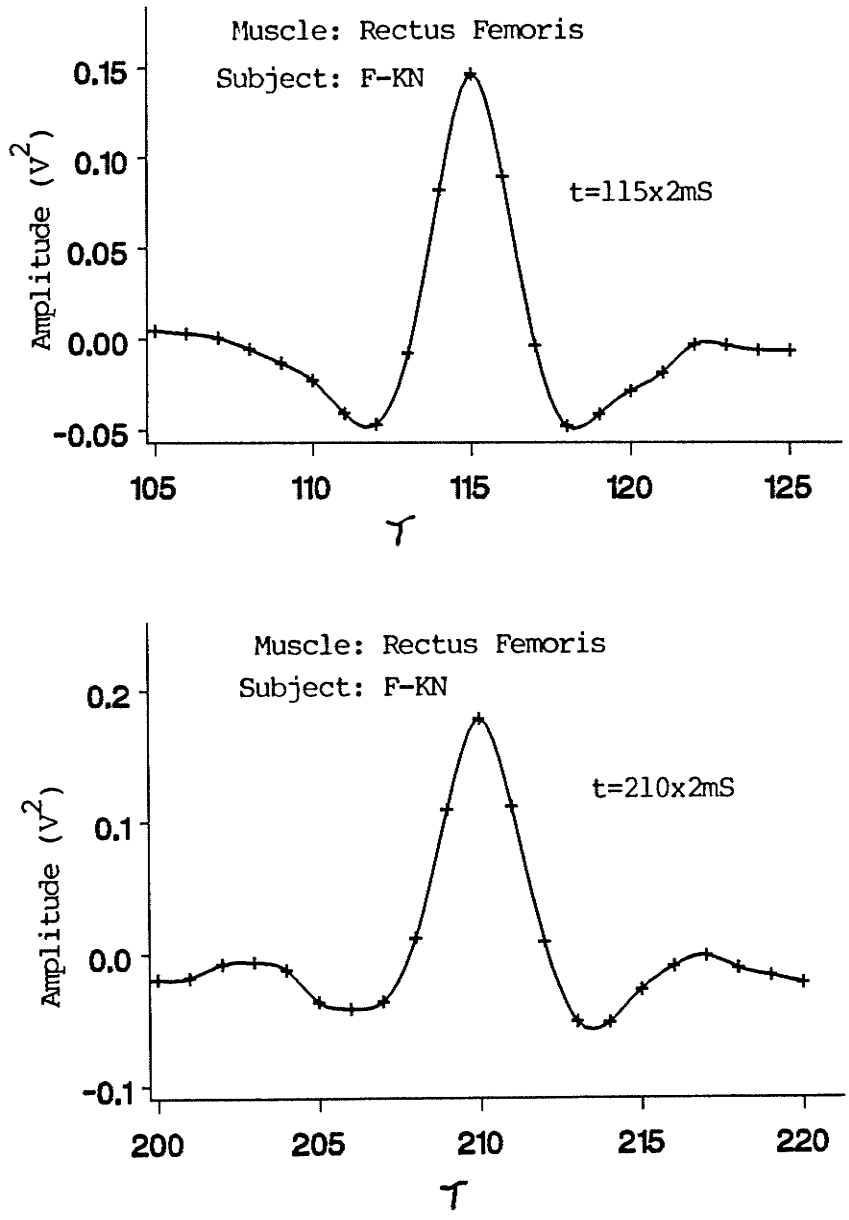


Figure 3.7: Autocorrelation of Experimental EMG

- t is the time instant(from the defined starting-point) at which the autocorrelation is calculated.
- to obtain τ in mS multiply by 2.

Besides the usual experimental errors the variance and autocorrelation functions exhibit statistical error due to the finite number of samples used to estimate them. In the case of the variance curve the 95% confidence interval is shown in Figure 3.8 as a function of N the sample size. The analysis is taken from Bendat[1971] and assumes that the data is generated by a Gaussian process which is valid in this situation. As can be seen for N>100 the estimated variance lies between 1.3 and 0.8 of the true value with a probability of 0.95.

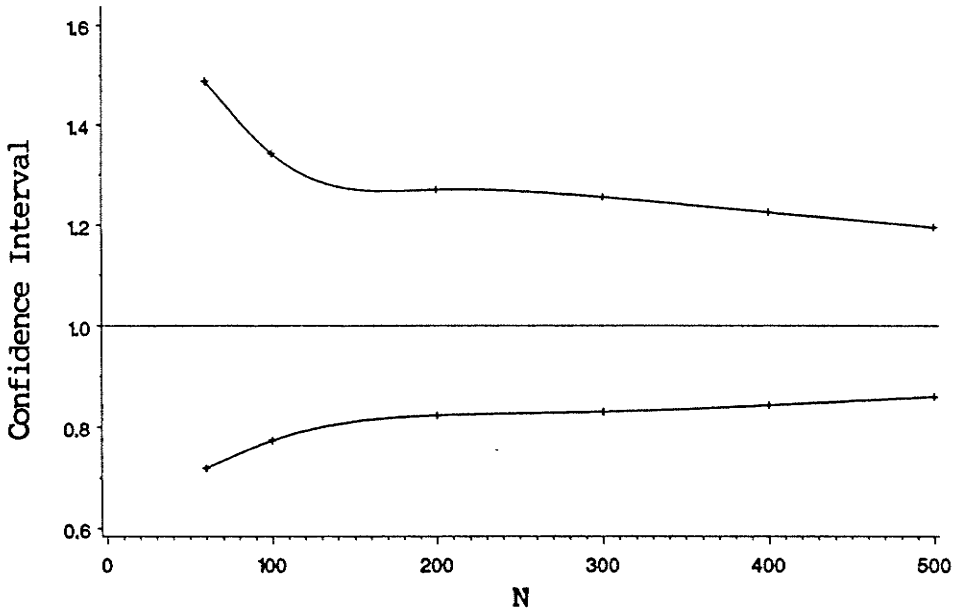


Figure 3.8: Confidence Interval of the EMG Variance
N - No. of trials included in the computation of the EMG variance

A complete analysis of the statistical error in estimating the autocorrelation is very complicated. Thus, an upper bound on the percent-mean-square-error was computed[Cooper, 1971] by using the following expression:

$$\text{Percent-Mean-square Error} \leqslant \frac{2}{N} \sum_{k=-M}^{M} R_X(k\Delta t) \ ,$$

where N is the total number of samples and R_X is the normalized autocorrelation which was taken to be the Parker's autocorrelation model combined with the corresponding variance function. The parameters of these functions were taken from the results described in the next chapter. The computed upper-bound-error for each ensemble is listed in Table 3.1. Theoretically, this error is monotonically decreasing when N is increasing. However, the two percent-errors at N=76 destroy the monotonicaity. This may be because data of these two ensembles were obtained from other muscles, and they have different estimates which consequently contribute to these deviated errors.

TABLE 3.1

Upper-Bound-Error of the Autocorrelation

	n(t)=ϭe1(t)	n(t)=ϭe2(t)
N	Error (%)	Error (%)
68	18.2	18.2
70	18.0	18.0
76	20.2	20.5
76	19.5	19.7
81	17.3	17.3
87	16.8	16.8
88	16.1	16.1
98	15.2	15.2
100	15.0	15.1
104	14.8	14.8
111	14.3	14.3
115	14.0	14.0
383	8.6	8.7
437	7.9	8.1
494	6.8	6.9

3.3.2 Curve-fitting

The EMG variance and autocorrelation were curve-fitted by the three functions mentioned in chapter 2.2, which are repeated below:

i. Variance functions:

$$\sigma^2_{e_1}(t)=K[1-\exp(-K_1 t)], \tag{3.7}$$

$$\sigma^2_{e_2}(t)=K[1-A_1\exp(-At)-B_1\exp(-Bt)] . \tag{3.8}$$

ii. Autocorrelation function:

$$R_{ww}(\tau)=\alpha^3\left(\frac{1}{\alpha^3}+\frac{|\tau|}{\alpha^2}-\frac{\tau^2}{\alpha}\right)\exp(-\alpha|\tau|), \tag{3.9}$$

where K_1, A, B and α are parameters to be estimated. All programs that estimated these parameters were written in the SAS language. Two procedures were used in each program, PROC NLIN and PROC GPLOT. PROC NLIN computed the best parameter for each function, and PROC GPLOT plotted the function and the corresponding curve.

Both of the variance functions were used to curve-fit the EMG variance. The scaling factor K of these two functions was taken to be the amplitude of the EMG variance at steady state. Program CFIT is written for function 3.7 and program DFIT for function 3.8.

One restriction when using function 3.8 is that constants A_1 and B_1 have to be related to parameter A and B so

that the first derivative of function 3.8 is zero when t=0. Thus, A_1 and B_1 can be expressed as follow:

$$A_1 = \frac{B}{A-B} \quad , \tag{3.10}$$

and

$$B_1 = \frac{-A}{A-B} \quad . \tag{3.11}$$

By combining function 3.9 with either function 3.7 or function 3.8, the following equation (equation 3.12) is used to curve-fit the experimentally obtained autocorrelation curves. These curve-fitting tasks are done by programs EDPARK and PARK2.

$$R_{ee}(\tau) = \delta e(t)\ \delta e(\tau+t)\ R_{ww}(\tau), \tag{3.12}$$

where $\delta e(t)$ is either $\delta e_1(t)$ or $\delta e_2(t)$. Finally, program PFIT curve-fits function 3.9 alone to the autocorrelation curves for comparison purpose.

Chapter IV

DISCUSSION AND APPLICATION

4.1 EXPERIMENTAL RESULTS AND DISCUSSION

As previously mentioned, fifteen data ensembles were obtained, eight small-ensembles from the biceps brachii, two small-ensembles from each of the rectus femoris and the deltoid, and three big-ensembles from the biceps brachii. In order to ensure that the data chosen from an ensemble was as consistent as possible, a mean-square-error threshold of 3% was used. If the error of a strain-gauge signal, when compared with the ensemble-average of strain-gauge signal, was below this threshold, the corresponding EMG signal was selected for the computation of variance and autocorrelation curves. However, since it is more difficult to control the rectus femoris and the deltoid muscles, the error threshold for these two muscles was raised to 20% to allow the use of more data. As a result, the variance and autocorrelation curves for these muscles show more fluctuation which consequently produces more error in the curve-fitted variances.

When curve-fitting the EMG variance with the variance function (2.7 and 2.8), there are parameters to be estimated in each function, parameter K_1 in $\delta_{e_1}^2$, and parameters A and B

in $\delta^2_{e_2}$. These parameters are listed in table 4.1, with the corresponding errors listed in table 4.2.

Table 4.1 shows that function $\delta^2_{e_2}(t)$ performs better than function $\delta^2_{e_1}(t)$, for it has relatively smaller error when curve-fitting the EMG variance. Function $\delta^2_{e_2}$ has less than 6.5% error while function $\delta^2_{e_1}$ has less than 8% error. One factor to account for this is that function $\delta^2_{e_1}$ cannot properly curve-fit the lower part of the EMG variance while function $\delta^2_{e_2}$ has an extra exponential term to overcome this problem. Table 4.1 also shows that both functions give better fits to the big-ensemble variance than they do to the small-ensemble ones; this is obviously because the larger ensembles produce a smoother EMG variance estimate. Results of curve-fitted EMG variance are shown in figure 4.1 and Appendix C.3.

For a given EMG e(t), the autocorrelation has been derived in equation 2.2, which states that:

$$R_{ee}(t+\tau)=n(t+\tau)n(t)R_{ww}(\tau),$$

where n(t) is either $\delta_{e_1}(t)$ or $\delta_{e_2}(t)$. Using the previously estimated parameters for $\delta^2_{e_1}(t)$ and $\delta^2_{e_2}(t)$, the above equation is used to curve-fit the experimentally obtained autocorrelations.

TABLE 4.1

Errors of the Best Estimates of $\sigma^2_{e_1}(t)$ and $\sigma^2_{e_2}(t)$

SMALL-ENSEMBLE			
		$\sigma^2_{e_1}(t)$	$\sigma^2_{e_2}(t)$
Subject	N	Error(%)	Error(%)
SM	87	3.88	3.87
AG	115	4.80	3.90
RK	104	2.90	2.60
XG	68	5.30	4.10
WG	88	1.16	0.78
BL	111	4.90	3.40
VN	70	3.20	2.23
KN	98	3.10	2.60
F-BL	76	7.87	6.33
D-BL	76	5.52	4.07
D-KN	100	3.24	2.73
F-KN	81	2.86	2.60
BIG-ENSEMBLE			
JJH	383	3.05	1.87
AAG	494	2.88	1.92
KKN	437	1.18	0.67

Notes:

(i). N-number of trial selected for calculating the EMG variance.

(ii). F-data obtained from the rectus femoris muscle.

(iii). B-data obtained from the deltoid muscle.

TABLE 4.2

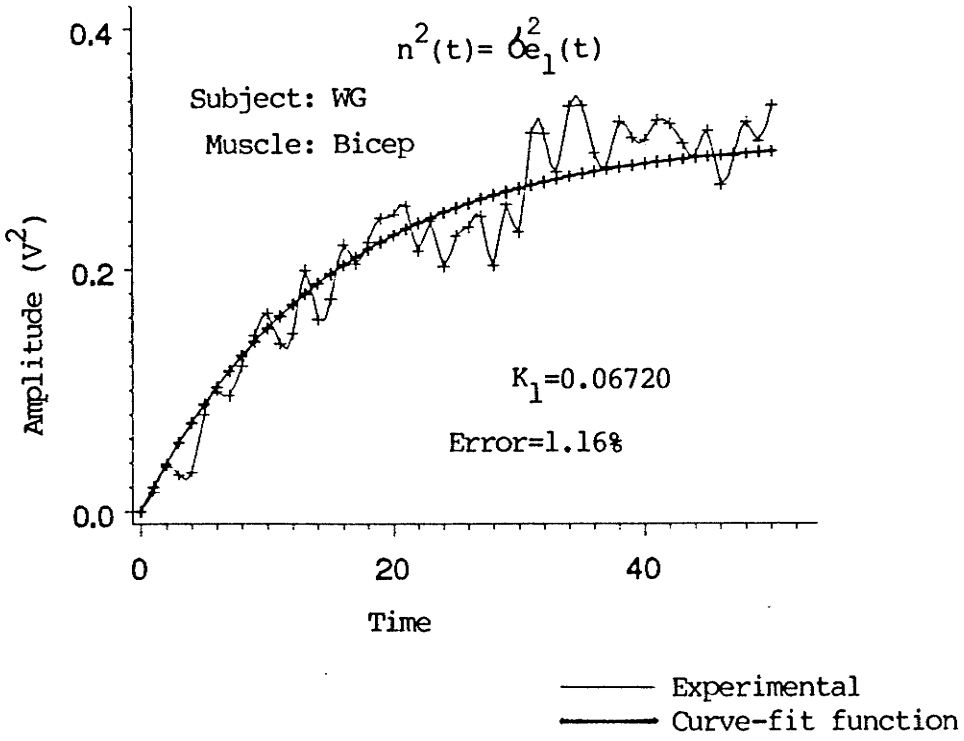
Estimated K_1, A & B, and the Calculated Starting-point

SMALL-ENSEMBLE					
	$\delta^2_{e_1}(t)$		$\delta^2_{e_2}(t)$		
Subject	Starting Point	K_1	Starting Point	A	B
SM	27	0.01281	27	0.55885	0.01317
AG	61	0.08389	56	0.20175	0.20434
RK	74	0.02541	70	0.05373	0.05227
XG	49	0.04890	47	0.09426	0.09415
WG	60	0.06718	55	0.11230	0.11229
BL	63	0.02651	43	0.03903	0.03827
VN	65	0.10551	65	0.21243	0.21887
KN	50	0.18014	49	0.30848	0.33314
F-BL	38	0.01972	38	0.03911	0.04086
D-BL	35	0.01595	25	0.03167	0.03389
D-KN	71	0.02852	49	0.03862	0.03860
F-KN	21	0.02744	10	0.06599	0.03722
BIG-ENSEMBLE					
JJH	10	0.01427	10	0.03179	0.03136
AAG	103	0.12341	99	0.10875	0.10963
KKN	47	0.05083	37	0.04262	0.07524

Notes:

(i). The starting-point is only a relative point for calculation and comparison convenience.

(ii). K_1, A and B are estimated parameters.



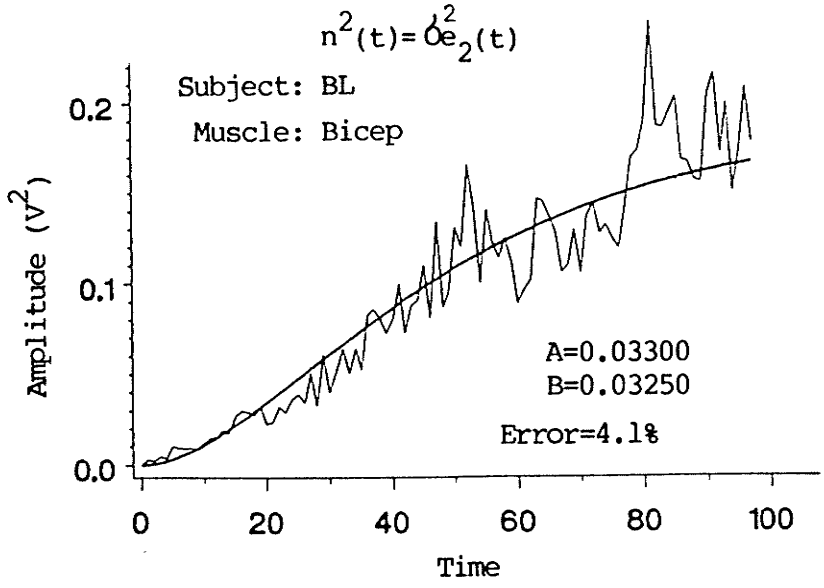


Figure 4.1: Curve-fitted EMG variance
- to obtain time in mS multiply by 2.

When this autocorrelation function (4.1) is used to curve-fit the experimental autocorrelations, only the parameter α of R_{ww} (2.4) needs to be estimated. It has been found that this parameter does not vary greatly for the set of autocorrelations of an ensemble. The set of α's was averaged. This averaged α for each ensemble is listed in Table 4.3 where errors of curve-fitting the autocorrelation curves with $n(t)=\acute{o}e_1(t)$ and the errors with $n(t)=\acute{o}e_2(t)$ are tabulated. Both variance functions give essentially the same error which ranges from 4% to 27%. Most of these errors are due to the fluctuation that occurs before and after the main-lobe of the experimentally obtained autocorrelations. Curve-fitted results are shown in figure 4.2 and Appendix C.4.

All of the curve-fitting done above is based on Kreifeldt's postulation which says that the EMG signal e(t) can be considered to be an amplitude modulated signal. By curve-fitting the experimentally obtained EMG variances and autocorrelations, it has been shown that the heuristically chosen variance functions, $\acute{o}^2e_1$ and $\acute{o}^2e_2$, and autocorrelation function, R_{ww}, performed quite satisfactorily in describing the EMG autocorrelation. Consequently, the experimental data supports the postulate that the EMG signal can be modelled as an amplitude modulated signal.

TABLE 4.3

Ensemble-Average of Parameter α of R_{ee} and Its Error

	SMALL-ENSEMBLE			
	$n(t)=\delta e_1(t)$		$n(t)=\delta e_2(t)$	
Subject	α	Error (%)	α	Error (%)
SM	0.71561	16.76	0.71007	15.71
AG	1.53717	22.07	1.53477	25.39
RK	1.36812	19.33	1.46004	19.60
XG	1.02436	21.26	1.02731	21.62
WG	1.20796	22.03	1.27619	28.41
BL	1.23108	18.68	1.27432	20.34
VN	1.20314	26.60	1.21400	26.19
KN	1.19102	21.18	1.25583	20.59
F-BL	0.50625	13.16	0.48734	11.86
D-BL	0.54908	13.88	0.53675	16.40
D-KN	0.95809	14.07	0.92723	12.38
F-KN	0.73305	17.38	0.72932	17.27
	BIG-ENSEMBLE			
JJH	0.55713	9.97	0.54167	5.20
AAG	0.82924	4.23	0.82192	4.91
KKN	0.59915	9.01	0.57151	11.75

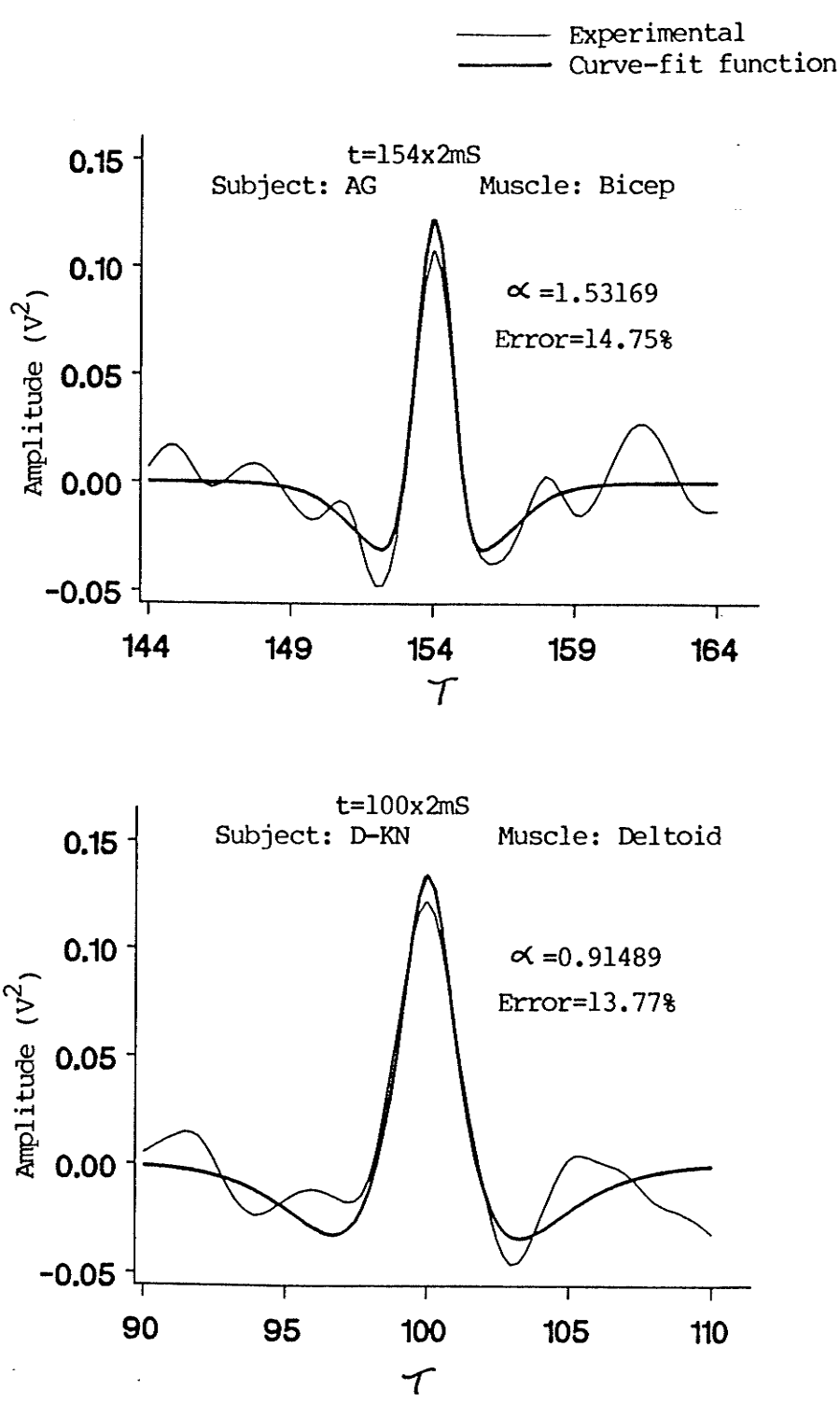


Figure 4.2: Curve-fitted Autocorrelation

- t is the time(from the defined starting-point) at which the autocorrelation is calculated.
- to obtain τ in mS multiply by 2.

4.2 EMG VARIANCE ESTIMATION

Up to now the estimates have been derived from the ensemble. Practically, the EMG variance has to be estimated from a single trial of data; therefore, two estimation schemes are evaluated here, they are the Midpoint Moving Average Estimation and the homomorphic filter.

4.2.1 Midpoint Moving Average Estimation (MMAE)

This algorithm ultilizes a window of constant weight where the variance is estimated at the midpoint. By moving this window point-by-point, squaring and averaging the EMG data within the window, an EMG variance estimate is obtained. The only parameter to be chosen is the window length. It has been found that the best estimate can be obtained by using approximately one-half of the fastest rise time of the time varying variance as the window length[Xiong, 1985]. The program used for this processor is listed in Appendix B.1.11. These estimated results were compared with the ensemble-average of the EMG variance. The error ranged from 2.5% to 105%. Out of 600 trials, 324 trials had less than 20% error, and 412 trials less than 30% error. A typical result of the MMAE is shown in figure 4.3.

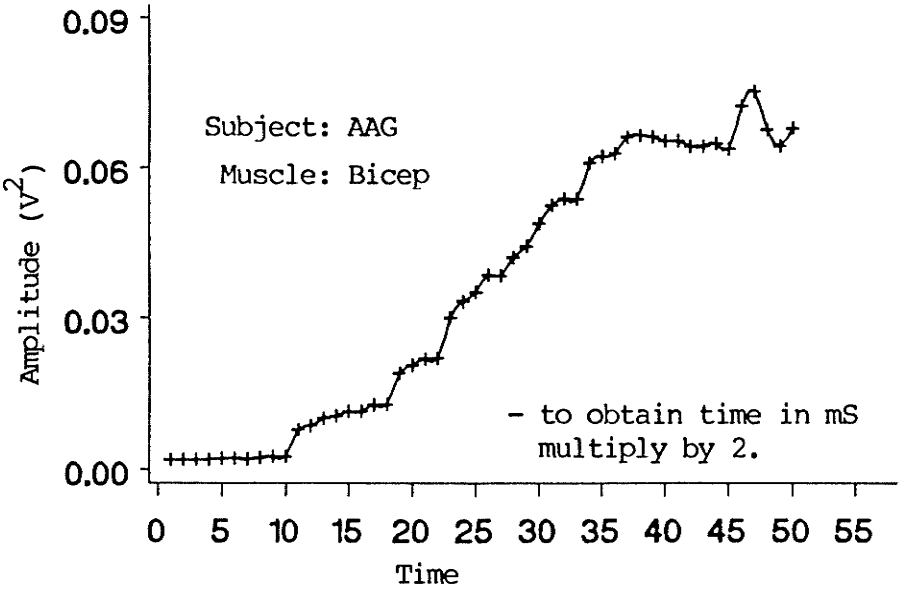


Figure 4.3: An Estimate Variance of the MMAE

4.2.2 Homomorphic Filter

Another possible processing algorithm used to estimate the EMG variance is a homomorpohic filter. Its block diagram is shown in figure 4.4.

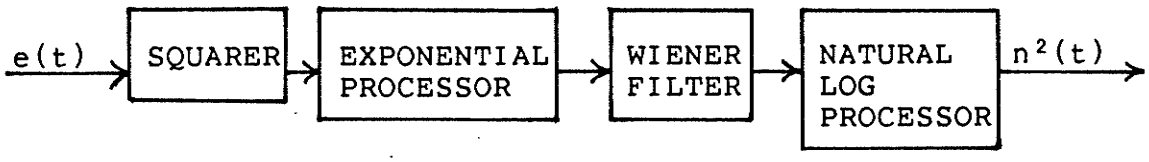


Figure 4.4: Block Diagram of the Homomorphic Filter

The square-processor squares the values of the EMG $e(t)$, and the log-processor converts the multiplicative characteristic of $e^2(t)$ to additive, i.e.

$$\ln[e^2(t)]=\ln[n^2(t)w^2(t)]$$

$$e'(t)=n'(t)+w'(t), \qquad (4.1)$$

so that a Wiener filter can be used to estimate the $n'(t)$. The design of this Wiener filter is illustrated in Appendix A. Its transfer function $H(s)$ is:

$$H(s)=K_3+\frac{K_4}{S+W_5} \qquad (4.2)$$

By taking the inverse Laplace transformation, one obtains the impulse response of the Wiener filter, which is

$$h(t)=K_3\delta(t)+K_4\exp(-W_5t). \qquad (4.3)$$

Program WIENER, which is listed in Appendix B.1.12, computed the variance $n^2(t)$ with $e(t)$ as the input. Program WIENEE compared the estimated variance with the ensemble-average of the EMG variance. The error of these estimated variances ranges from 3.42% to 157.22%. Out of 600 trials, 112 trials have less than 30% error and 56 trials have less than 20% error.

Chapter V

CONCLUSION

The primary objective of this thesis was to investigate the dynamics of the EMG signal characteristics. This was accomplished by asking subjects perform a two level tracking study with the EMG variance and autocorrelation function computed from the resultant ensemble.

Comparison of the experimentally obtained autocorrelation function with that derived from an amplitude modulated model of EMG signal generation shows good agreement; the mean-square-errors range from 4.2% to 26.6%. Thus the experimental study supports the model of signal generation. Further the study shows that the autocorrelation function does not change in form nor does the parameter α change to any great extent as the EMG signal characteristics evolve with time, i.e., as the subject muscle goes from a relaxed state to a contracted state. Therefore, the signals' power density spectrum does not change in form, only the power level increases as shown by the increasing variance. Though the experiment considered only a step change in target level, this could be considered to be an extreme case of muscle contraction and therefore the general conclusions that the signal generation is well modelled as an amplitude modulated

process and that the power density spectrum does not change in shape should apply to the more general case.

In practical application to prosthetic control the variance needs to be estimated from a single member of the ensemble. Two estimation schemes, the Midpoint Moving Average Estimator and the homomorphic filter were evaluated. The experimentally determined ensemble averages were taken as the true time varying variance. Results were mixed; out of 600 trials the MMAE had 324 estimates with error of less than 20% while the homomorphic filter had only 56 estimates. The poorer performance of the homomorphic filter can be partly explained in that a model for the signal and noise power density spectrum at the log processor's output was not available. These spectra, necessary for the Wiener filter design, were chosen to be simply low pass processes. However, given the derivation of the Wiener filter it is not felt that even with better spectra models the homomorphic filter would improve on the variance estimate.

With regard to future research there are several avenues which may be explored. Different skeletal muscles may be investigated. In general though it would be expected that the findings of this research would hold, only the model parameters would change. The present study was confined to isometric studies; an obvious research extension would be to consider EMG signal generation under non-isometric contraction. Finally although effort was made during the ex-

periment to ensure that each EMG ensemble member was produced under the same conditions, the results of the single trial estimates show a wide range in error suggesting that conditions did change. This change may be caused by the unexpected movements of the subject during data acquisition. Therefore, the experimentaly paradigm should be further investigated.

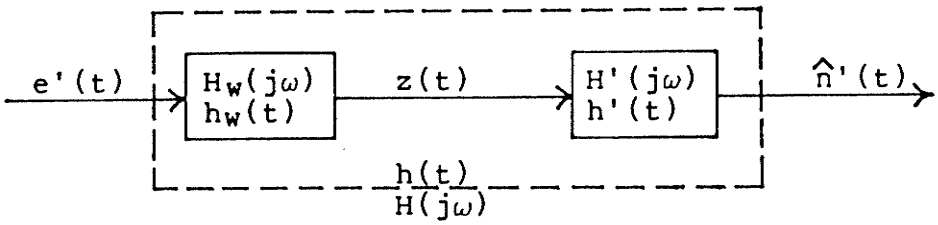
Appendix A

WIENER FILTER

The derivation here is based on the reference [Van Tree, 1968]. First, consider the following EMG signal e'(t).

$$e'(t)=n'(t)+w'(t),$$

where e'(t) is the EMG signal that has been passed through a square-processor and a log-processor, n'(t) is the signal to be estimated and w'(t) is the noise. The following diagram shows the block diagram of a Wiener filter which is used to estimate the n'(t).



where $\hat{n}'(t)$ is the estimate of n'(t). The transfer functions, $h_w(t)$ and h'(t), depend on the spectral properties of n'(t) and w'(t). Therefore, assume that $S_{nn}(\omega)$ and $S_{ww}(\omega)$ are the power spectra of n'(t) and w'(t) respectively. Previous study has shown that the power spectrum of w'(t) can be expressed as follows [Scott, 1967] and [Shwedyk, 1973]:

$$S_{ww}(\omega) = \frac{K_2}{(1+ \omega^2/ \omega_m^2)} \quad , \qquad (A.1)$$

where $\omega_m = 2\pi(100Hz)$. Also, since the n'(t) is a low frequency signal, its power spectrum can be assumed to be the following:

$$S_{nn}(\omega) = \frac{K_1}{(1+ \omega^2/ \omega_n^2)} \quad , \qquad (A.2)$$

where $\omega_n = 2\pi(20Hz)$. Thus, the spectrum $S_{ee}(\omega)$ is given by

$$S_{ee}(\omega) = S_{nn}(\omega) + S_{ww}(\omega) + S_{nw}(\omega) + S_{wn}(\omega),$$

where $S_{nw}(\omega)$ and $S_{wn}(\omega)$ are only constants. They are neglected in subsequent derivation because they do not affect the design of the system response. Therefore,

$$S_{ee}(\omega) = S_{nn}(\omega) + S_{ww}(\omega),$$

$$= \frac{K_1}{(1+ \omega^2/\omega_n^2)} + \frac{K_2}{(1+ \omega^2/\omega_m^2)},$$

$$= \frac{K_1 + K_2 + \omega^2 (K_1/\omega_m^2 + K_2/\omega_n^2)}{(1+\omega^2/\omega_n^2)(1+\omega^2/\omega_m^2)}---, \qquad (A.3)$$

By spectral factorization, this can be rewritten as

$$S_{ee}(\omega) = [G^+(j\omega)][G^+(j\omega)]^* \qquad (A.5)$$

where

$$[G^+(j\omega)] = \frac{K + j\omega L}{(1+j\omega/\omega_n)(1+j\omega/\omega_m)}---, \qquad (A.6)$$

$$[G^{+}(j\omega)] = \frac{K - j\,L}{(1-j\omega/\omega_n)(1+j\omega/\omega_m)}, \qquad (A.7)$$

$$K = (K_1+K_2)^{1/2},$$

and

$$L = (K_1/\omega_m^2+K_2/\omega_n^2)^{1/2}.$$

Since the transfer function $H_w(j\omega)$ of the Wiener filter is defined as:

$$H_w(j\omega) = \frac{1}{[G^{+}(j\omega)]},$$

Therefore,

$$H_w(j\omega) = \frac{(1+j\omega/\omega_n)(1+j\omega/\omega_m)}{(K+j\omega L)}. \qquad (A.8)$$

Further, for the transfer function h'(t), we have to consider the cross-correlation $S_{en}(\omega)$ which is:

$$S_{en}(\omega) = S_{nn}(\omega) + S_{wn}(\omega),$$

$$= S_{nn}(\omega),$$

$$= \frac{K_1}{(1+\omega^2/\omega_n^2)}. \qquad (A.10)$$

The previous block diagram shows that

$$S_{nz}(j\omega) = S_{en}(\omega)\ H_w^{*}(j\omega),$$

$$= \frac{S_{en}(\omega)}{[G^{+}(j\omega)]^{*}} . \qquad (A.11)$$

Thus,
$$S_{nz}(j\omega) = \frac{K_1(1-j\omega/\omega_n)(1-j\omega/\omega_m)}{(1+\omega^2/\omega_n^2)(K-j\omega L)} ,$$

$$= \frac{K_1(1-j\omega/\omega_m)}{(1+j\omega/\omega_n)(K-j\omega L)} . \qquad (A.12)$$

By partial fraction, we obtain

$$S_{nz}(j\omega) = \frac{A}{(1+j\omega/\omega_n)} + \frac{B}{(K-j\omega L)} , \qquad (A.13)$$

where

$$A = \frac{K_1(1-j\omega/\omega_m)}{(K-j\omega L)},$$

with $\omega = j\omega_n$, it becomes

$$A = \frac{K_1(1+\omega_n/\omega_m)}{K+\omega_n L}, \qquad (A.14)$$

and similarly,

$$B = \frac{K_1(1-K/(L\,\omega_m))}{(1+K/(L\,\omega_n))} . \qquad (A.15)$$

Because the second term of equation A.13 has a pole on the R.H.P., it has to be discarded in order to make the filter stable. Thus, we define

$$[Snz(j\omega)]_+ = \frac{A}{(1+j\omega/\omega_n)},$$

and this is also the transfer function of $H'(j\omega)$. Therefore,

$$H'(j\omega) = \frac{A}{(1+j\omega/\omega_n)} \qquad (A.16)$$

Hence, the overall transfer function of the Wiener filter can be defined as follow:

$$H(j\omega) = H_w(j\omega)H'(j\omega). \qquad (A.17)$$

Substitute equations A.8 and A.16 into A.17, we obtain

$$H(j\omega) = \frac{A(1+j\omega/\omega_m)}{K+j\omega L},$$

let $s=j\omega$,

$$H(s) = \frac{A(1+s/\omega_m)}{K+sL}$$

$$= \frac{A}{L} + \frac{A(1-K/L\omega_m)}{L(s+K/L)}.$$

Identify

$$K_3 = \frac{A}{L},$$

$$K_4 = \frac{A(1-K/(L\omega_m))}{L},$$

and $$\omega_5 = \frac{K}{L} ,$$

therefore, the transfer function of the Wiener filter becomes,

$$H(s) = K_3 + \frac{K_4}{(s + \omega_5)},$$

or $$h(t) = K_3\delta(t) + K_4\exp(-\omega_5 t) , \qquad (A.18)$$

which is illustrated in the following block diagram:

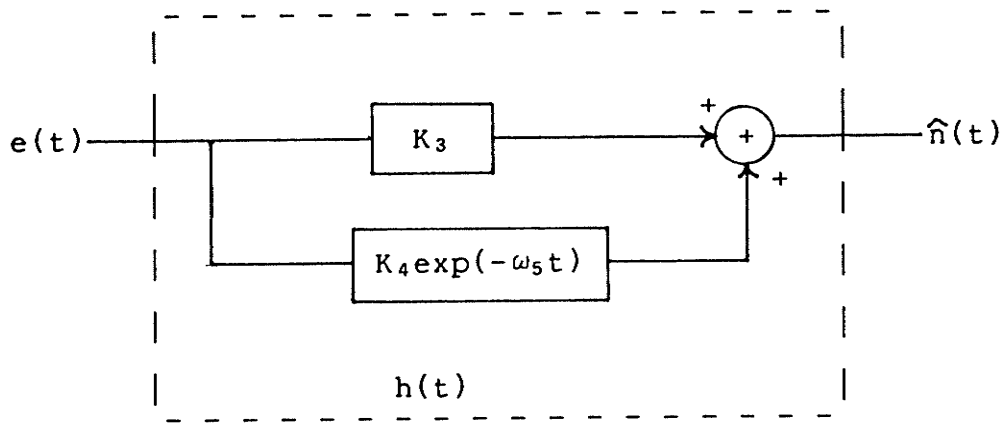


Figure A.1: Wiener Filter

Appendix B

COMPUTER PROGRAM

B.1 PROGRAMS USED IN MICRO 11/23

B.1.1 Program SORT.FTN

```
      PROGRAM SORT
      CHARACTER*6 FNAME
      CHARACTER*8 EMFILE,SGFILE
      PARAMETER (FNAME='K2',SGFILE='KSG3',EMFILE='KEM3',
     C           LAST=5000)
      INTEGER*2 SG(LAST),EMG(LAST)
      OPEN (UNIT=3,FILE=FNAME,READONLY,STATUS='OLD',BLANK='ZERO')
            READ(3,10) (SG(I),EMG(I),I=1,LAST)
 10         FORMAT(8(2I4))
      CLOSE (UNIT=3,DISPOSE='KEEP')
      OPEN (UNIT=4,NAME=SGFILE,STATUS='NEW',BLANK='ZERO',
     C      BLOCKSIZE=42)
            WRITE(4,20) (SG(I),I=1,LAST)
 20         FORMAT(X,16I4)
      CLOSE (UNIT=4,DISPOSE='KEEP')
      OPEN (UNIT=1,NAME=EMFILE,STATUS='NEW',BLANK='ZERO',
     C      BLOCKSIZE=42)
            WRITE(1,30) (EMG(I),I=1,LAST)
 30         FORMAT(X,16I4)
      CLOSE (UNIT=1,DISPOSE='KEEP')
      STOP
      END
```

B.1.2 Program MIDPOINT.FTN

```
 1.        CHARACTER*8 FNAME(100)
 2.        INTEGER*2 COUNT1,COUNT2,COUNT3,I,X(5000),SAMPLE,MID
 3.        PARAMETER (NFILE=50,LAST=5000,SAMPLE=20)
 4.        REAL*4 SUM,MAX(NFILE),ZERO(NFILE),MAX1,ZERO1,DUM
 5.        DATA MAX1/0.0/,ZERO1/0.0/,MAX/NFILE*0.0/,ZERO/NFILE*5000.0/
 6.  *
 7.              FNAME(1)='KSG1'
 8.  C                 TO
 9.             FNAME(50)='KSG50'
10.  *
11.        DO 50 COUNT1=1,NFILE
12.           OPEN (UNIT=3,FILE=FNAME(COUNT1),STATUS='OLD')
13.                READ (3,10) (X(I),I=1,LAST)
14.  10            FORMAT (X,16I4)
15.           CLOSE (UNIT=3,DISPOSE='KEEP')
16.  *
17.           DO 40 COUNT2=1,200
18.              SUM=0.0
19.              I=(COUNT2-1)*SAMPLE
20.              DO 20 COUNT3=1,SAMPLE
21.  20             SUM=SUM+X(I+COUNT3)/SAMPLE
22.              IF (SUM .LT. MAX(COUNT1)) GO TO 30
23.                  MAX(COUNT1)=SUM
24.  30          IF (SUM .GT. ZERO(COUNT1)) GO TO 40
25.                  ZERO(COUNT1)=SUM
26.  40       CONTINUE
27.  50    CONTINUE
28.  *
29.        DO 60 COUNT3=1,NFILE
30.           MAX1=MAX1+MAX(COUNT3)/(NFILE)
31.           ZERO1=ZERO1+ZERO(COUNT3)/(NFILE)
32.  60    CONTINUE
33.  *
34.        DUM=((MAX1-ZERO1)*0.5)+ZERO1
35.        MID=IINT(DUM)
36.  *
37.        OPEN (UNIT=3,NAME='KMPONT',STATUS='NEW')
38.             WRITE(3,70) (ZERO(I),MAX(I),I=1,NFILE)
39.  70         FORMAT (5X,F7.2,5X,F7.2)
40.             WRITE(3,80) MID
41.  80         FORMAT (X,'THE AVERAGE MIDPOINT IS',I6)
42.        CLOSE (UNIT=3,DISPOSE='KEEP')
43.        STOP
44.        END
```

B.1.3 Program STARTER.FTN

```
 1.        PARAMETER (NFILE=50,LAST=5000,MIDDLE=1547,DISTAN=1200)
 2.        CHARACTER*8 FNAME(NFILE)
 3.        INTEGER*2 COUNT1,SG(LAST),I,NSAM,DUM,BACK,
 4.       C          START(600),A
 5.        DATA NSAM/0/,DUM/0/,BACK/100/
 6. *
 7.                FNAME(1)='KSG1'
 8. C                 TO
 9.               FNAME(50)='KSG50'
10. *
11.        DO 200 COUNT1=1,NFILE
12. *
13.           OPEN (UNIT=2,NAME=FNAME(COUNT1),STATUS='OLD')
14.                READ (2,210) (SG(I),I=1,LAST)
15. 210            FORMAT (X,16I4)
16.           CLOSE (UNIT=2,DISPOSE='KEEP')
17. *
18.           I=0
19. 220       I=I+10
20.           IF ((I) .GE. LAST) GO TO 200
21.           IF (SG(I) .LT. MIDDLE) GO TO 220
22.           I=I-11
23. 230       I=I+1
24.           IF (SG(I) .LT. MIDDLE) GO TO 230
25.           NSAM=NSAM+1
26.           DUM=DUM+1
27.           A=I-BACK
28.           I=I+DISTAN
29.           START(DUM)=A
30.           GO TO 220
31. 200    CONTINUE
32. *
33.        OPEN (UNIT=3,NAME='KSTARTER',STATUS='NEW')
34.             WRITE(3,281) (START(I),I=1,DUM)
35. 281          FORMAT(X,3I8)
36.        CLOSE (UNIT=3,DISPOSE='KEEP')
37.        STOP
38.        END
```

B.1.4 Program MINSG.FTN

```
      PROGRAM MINSG
      PARAMETER (NFILE=40,LAST=5000,NUM=600,NSAM=3,JUMP=20)
      CHARACTER*8 FNAME(NFILE)
      INTEGER*2 COUNT1,COUNT2,COUNT3,SG(LAST),I,NUM1,
     C          Z,START(600),AVE(NUM),COUNT4
      INTEGER*4 SUM(NUM)
      REAL Y
      DATA NUM1/0/,AVE/NUM*0/,SUM/NUM*0/
*
              FNAME(1)='KSG1'
     C                TO
             FNAME(40)='KSG40'
*
      OPEN (UNIT=3,NAME='KSTARTER',STATUS='OLD')
           READ(3,*) (START(I),I=1,NSAM*NFILE)
*81        FORMAT(X,3I8)
      CLOSE (UNIT=3,DISPOSE='KEEP')
*
      DO 270 COUNT=1,NFILE
*
          OPEN (UNIT=2,FILE=FNAME(COUNT),STATUS='OLD')
               READ(2,280) (SG(I),I=1,LAST)
280            FORMAT(X,16I4)
          CLOSE (UNIT=2,DISPOSE='KEEP')
*
          DO 240 COUNT2=1,NSAM
             I=START((COUNT-1)*NSAM+COUNT2)-1+NUM+JUMP
             IF (I.GT.LAST .OR. I.LT.0) GO TO 240
             I=I-NUM
             NUM1=NUM1+1
             DO 250 COUNT3=1,NUM
                SUM(COUNT3)=SUM(COUNT3)+SG(I+COUNT3)
250          CONTINUE
240       CONTINUE
270   CONTINUE
      DO 290 COUNT4=1,NUM
         Y=SUM(COUNT4)/NUM1
         Z=IINT(Y)
         AVE(COUNT4)=Z
290   CONTINUE
*
      OPEN (UNIT=2,NAME='KMINSG',STATUS='NEW')
             WRITE(2,260) (AVE(I),I=1,NUM)
260          FORMAT(X,16I4)
             CALL PLOT(AVE,NUM,2)
      CLOSE (UNIT=2,DISPOSE='KEEP')
*
      STOP
      END
```

B.1.5 Program VARSG.FTN

```
1.        PROGRAM VARSG
2.        PARAMETER (NFILE=40,LAST=5000,NSAM=3,JUMP=50,NUM=100)
3.        INTEGER*2 START(600),COUNT1,COUNT2,COUNT3,
4.       C          SG(LAST),MEAN(NUM)
5.        REAL*4 SUM,PRO,DELTA,DUM,DDUM,NORM
6.        REAL*4 VAR(600)
7.        CHARACTER*8 FNAME(200)
8.        DATA VAR/600*-99.0/,NORM/0.0/
9.  *
10.               FNAME(1)='KSG1'
11. C                    TO
12.               FNAME(40)='KSG40'
13. *
14.       OPEN (UNIT=2,FILE='KMINSG',STATUS='OLD')
15.            READ(2,901) (MEAN(I),I=1,NUM)
16. 901        FORMAT(X,16I4)
17.       CLOSE (UNIT=2,DISPOSE='KEEP')
18. *
19.       DO 940 COUNT1=JUMP,NUM
20.          DUM=MEAN(COUNT1)
21.          DDUM=(DUM/(NUM-JUMP))*DUM
22.          NORM=NORM+DDUM
23. 940   CONTINUE
24.       NORM=SQRT(NORM)
25. *
26.       OPEN (UNIT=2,FILE='KSTARTER',STATUS='OLD')
27.            READ(2,*) (START(I),I=1,NFILE*NSAM)
28.       CLOSE (UNIT=2,DISPOSE='KEEP')
29. *
30.       DO 910 COUNT1=1,NFILE
31.          OPEN (UNIT=2,FILE=FNAME(COUNT1),STATUS='OLD')
32.               READ(2,911) (SG(I),I=1,LAST)
33. 911           FORMAT(X,16I4)
34.          CLOSE (UNIT=2,DISPOSE='KEEP')
35. *
36.          DO 920 COUNT2=1,NSAM
37.             SUM=0.0
38.             I1=(COUNT1-1)*NSAM+COUNT2
39.             I=START(I1)-1+JUMP
40.             IF ((I+NUM).GT.LAST .OR. I.LT.0) GO TO 920
41.             DO 930 COUNT3=JUMP,NUM
42.                DELTA=(MEAN(COUNT3)-SG(I+COUNT3))
43.                PRO=(DELTA/(NUM-JUMP))*DELTA
44.                SUM=SUM+PRO
45. 930         CONTINUE
46.             VAR(I1)=(SQRT(SUM)/NORM)*100.0
47. 920      CONTINUE
48. 910   CONTINUE
49. *
50.       OPEN (UNIT=2,NAME='KVARSG',STATUS='NEW')
51.            WRITE(2,*) (VAR(I),I=1,NSAM*NFILE)
52.            WRITE(2,921)
53. 921        FORMAT(X,'The above data are Root-Mean-Square-Error ',
54.      c                 'with unit of %')
55.       CLOSE (UNIT=2,DISPOSE='KEEP')
56.       STOP
57.       END
```

B.1.6 Program SCALE.FTN

```
      PROGRAM SCALE
      CHARACTER*9 OUT(100)
      CHARACTER*8 EMFILE(100)
      PARAMETER (LAST=5000,NFILE=50)
      INTEGER*2 COUNT1,COUNT2,EMG(LAST)
      REAL*4 SEMG(LAST)
C
           EMFILE(1)='AAEM1'
                     TO
          EMFILE(100)='AAEM100'
C
               OUT(1)='AAS1'
                     TO
              OUT(100)='AAS100'
C
      DO 400 COUNT1=1,NFILE
         OPEN (UNIT=3,FILE=EMFILE(COUNT1),STATUS='OLD')
              READ(3,410) (EMG(J),J=1,LAST)
410           FORMAT (X,16I4)
         CLOSE (UNIT=3,DISPOSE='delete')
               DO 420 COUNT2=1,LAST
                  SEMG(COUNT2)=(EMG(COUNT2)/4096.0)*2.0-1.0
420            CONTINUE
         OPEN (UNIT=3,NAME=OUT(COUNT1),STATUS='NEW')
              WRITE(3,430) (SEMG(J),J=1,LAST,1)
430           FORMAT(5E15.7)
         CLOSE (UNIT=3,DISPOSE='KEEP')
400   CONTINUE
         STOP
         END
```

B.1.7 Program OSEMG.FTN

```
 1.        PROGRAM OSEMG
 2.        PARAMETER (NFILE=50,LAST=5000)
 3.        INTEGER*2 COUNT1,COUNT2
 4.        REAL*4 MEAN,EMG(LAST)
 5.        DATA MEAN/0.0/
 6.        CHARACTER*9 EMFILE(50),OUT(50)
 7. C
 8.           EMFILE(1)='KKSEM1'
 9.                    TO
10.           EMFILE(50)='KKSEM50'
11. C
12.               OUT(1)='KKOEM1'
13.                  TO
14.              OUT(50)='KKOEM50'
15. C
16.        DO 520 COUNT1=1,NFILE
17.           OPEN (UNIT=4,FILE=EMFILE(COUNT1),STATUS='OLD')
18.                READ(4,500) (EMG(I),I=1,LAST)
19. 500            FORMAT(5E15.7)
20.           CLOSE(UNIT=4,DISPOSE='KEEP')
21. *
22.           DO 510 COUNT2=1,LAST
23. 510          MEAN=MEAN+EMG(COUNT2)
24. 520    CONTINUE
25.        MEAN=MEAN/(NFILE*LAST)
26. *
27.        OPEN (UNIT=4,NAME='OSEMG',STATUS='NEW')
28.             WRITE(4,530) MEAN
29. 530          FORMAT(X,E15.7)
30.        CLOSE(UNIT=4,DISPOSE='KEEP')
31. *
32.        DO 540 COUNT1=1,NFILE
33.           OPEN (UNIT=4,FILE=EMFILE(COUNT1),STATUS='OLD')
34.                READ(4,550) (EMG(I),I=1,LAST)
35. 550            FORMAT(5E15.7)
36.           CLOSE(UNIT=4,DISPOSE='DELETE')
37. *
38.           DO 560 COUNT3=1,LAST
39. 560          EMG(COUNT3)=EMG(COUNT3)-MEAN
40. *
41.           OPEN (UNIT=1,NAME=OUT(COUNT1),STATUS='NEW')
42.                WRITE(1,570) (EMG(I),I=1,LAST)
43. 570            FORMAT(5E15.7)
44.           CLOSE (UNIT=1,DISPOSE='KEEP')
45. 540    CONTINUE
46.        STOP
47.        END
```

B.1.8 Program MEMG.FTN

```
1.       PARAMETER (NFILE=40,LENGTH=500,JUMP=50,RATIO=20.0,
2.      C           LAST=5000)
3.       INTEGER*2 COUNT1,START(600),
4.      C          COUNT2,COUNT4,M,I,I2
5.       REAL*4 EMG(LAST),MEMG(LENGTH),Z,Z2,VAR,VARSG(120),
6.      C       PMEMG(LENGTH)
7.       CHARACTER*9 FNAME(100)
8.       DATA MEMG/LENGTH*0.0/,M/0/
9.  *
10.      OPEN (UNIT=2,FILE='KSTARTER',STATUS='OLD')
11.           READ(2,*) (START(I),I=1,NFILE*3)
12.      CLOSE (UNIT=2,DISPOSE='KEEP')
13. *
14.              FNAME(1)='KOEM1'
15. C                 TO
16.             FNAME(40)='KOEM40'
17. *
18.      OPEN (UNIT=3,FILE='KVARSG',STATUS='OLD')
19.           READ(3,*) (VARSG(I),I=1,NFILE*3)
20.      CLOSE (UNIT=3,DISPOSE='KEEP')
21. *
22.      DO 640 COUNT1=1,NFILE
23.         OPEN (UNIT=2,FILE=FNAME(COUNT1),STATUS='OLD')
24.              READ(2,*) (EMG(I),I=1,LAST)
25.         CLOSE (UNIT=2,DISPOSE='KEEP')
26. *
27.         DO 650 COUNT2=1,3
28.           I2=COUNT2+(COUNT1-1)*3
29.            I=START(I2)-1+JUMP
30.            Z=VARSG(I2)
31.            IF ((Z .LE. RATIO) .AND. (Z.GE.0.0)) THEN
32.               M=M+1
33.               DO 670 COUNT4=1,LENGTH
34.                  Z2=EMG(I+COUNT4)
35. 670              MEMG(COUNT4)=MEMG(COUNT4)+Z2*Z2
36.            END IF
37. 650     CONTINUE
38. 640  CONTINUE
39. *
40.      DO 622 COUNT1=1,LENGTH
41. 622     MEMG(COUNT1)=MEMG(COUNT1)/M
42. *
43.      OPEN (UNIT=4,NAME='KMEMG',STATUS='NEW')
44.            WRITE(4,684) M
45. 684        FORMAT(' The total number of sample that satisfied the
46.     C RATIO criterion is',I4,'.')
47.            WRITE (4,681) RATIO,JUMP
48. 681        FORMAT(' This is the variance of EMG with ',
49.     C          'RMS-Error <',F7.3,'%', 'and JUMP=',I3,'.')
50.            WRITE(4,680) (MEMG(I),I=1,LENGTH)
51. 680        FORMAT (5E15.7)
52.      CALL PLOT(MEMG,LENGTH,1)
53.      CLOSE (UNIT=4,DISPOSE='KEEP')
54.      STOP
55.      END
```

B.1.9 Program ACEMG.FTN

```
 1.       PARAMETER (IDELAY=290,NFILE=40,NSAM=3,IBEGIN=261,IANSAM=320,
 2.      C           RATIO=3.0,LAST=5000,JUMP=50)
 3.       INTEGER*2 START(120),I,I1,M,K(IANSAM),
 4.      C          COUNT1,COUNT2,COUNT3
 5.       REAL*4    VAR,VARSG1(120),Z,EMG(LAST),X,ACEMG(IANSAM),TAU
 6.       DATA      ACEMG/IANSAM*0.0/,M/0/
 7.       CHARACTER*9  EMFILE(40)
 8. *
 9.            EMFILE(1)='KOEM1'
10. C                  TO
11.            EMFILE(40)='KOEM40'
12. *
13.       OPEN (UNIT=2,FILE='KSTARTER',STATUS='OLD')
14.            READ(2,*) (START(I),I=1,NFILE*3)
15.       CLOSE (UNIT=2,DISPOSE='KEEP')
16. *
17.       OPEN (UNIT=3,FILE='KVARSG',STATUS='OLD')
18.            READ(3,*) (VARSG1(I),I=1,3*NFILE)
19.       CLOSE (UNIT=3,DISPOSE='KEEP')
20. *
21.       DO 700 COUNT1=1,NFILE
22.          OPEN (UNIT=2,FILE=EMFILE(COUNT1),STATUS='OLD')
23.               READ(2,*)(EMG(I),I=1,LAST)
24.          CLOSE (UNIT=2,DISPOSE='KEEP')
25. *
26.          DO 710 COUNT2=1,NSAM
27.             I1=(COUNT1-1)*NSAM+COUNT2
28.             I=START(I1)-1+JUMP
29.             Z=VARSG1(I1)
30.             IF ((Z .LE. RATIO).AND.(Z.GE.0.0)) THEN
31.                 M=M+1
32.                 TAU=EMG(I+IDELAY)
33.                 DO 720 COUNT3=IBEGIN,IANSAM
34.                    X=EMG(I+COUNT3)*TAU
35.                    ACEMG(COUNT3)=ACEMG(COUNT3)+X
36. 720             CONTINUE
37.             END IF
38. 710       CONTINUE
39. 700    CONTINUE
40. *
41.        DO 730 COUNT1=IBEGIN,IANSAM
42. 730       ACEMG(COUNT1)=(ACEMG(COUNT1))/M
43. *
44.        OPEN (UNIT=3,NAME='KACEMG',STATUS='NEW')
45.             WRITE(3,702) IDELAY,RATIO,JUMP
46. 702          FORMAT(X,'This is the autocorrelation of EMG
47.       c with DELAY= ',I3,'*2mS, RATIO=',F5.2,'and JUMP=',I3)
48.             WRITE(3,703) IANSAM,M
49. 703          FORMAT(X,'It consists of ',I3,' samples-length and M=',I3)
50.             CALL PLOT(ACEMG,IBEGIN,IANSAM,1)
51.        CLOSE (UNIT=3,DISPOSE='KEEP')
52.        STOP
53.        END
```

B.1.10 Program MMAE.FTN

```
 1.        PROGRAM MMAE
 2.        PARAMETER (L=12,LEN=90,NFILE=100,LAST=5000,RATIO=3.0)
 3.        INTEGER I,K,K2,I1,K3,K4,START(600),KK,NNFILE,M,NUM
 4.        REAL*4 EMG(LAST),VAR(600)
 5.        REAL*4 AVE(LEN),DUM,DUM1,DUM2,ME(LEN)
 6.        CHARACTER*9 FNAME(100),OUTPUT(2)
 7.        DATA AVE/LEN*0.0/,NUM/0/
 8. C
 9.                FNAME(1)='AAM1'
10. C                   TO
11.             FNAME(100)='AAM100'
12. C
13.        OPEN (UNIT=3,NAME='AASTARTER',STATUS='OLD')
14.           READ(3,*) (START(I),I=1,600)
15.        CLOSE(UNIT=3,DISPOSE='KEEP')
16. C
17.        OPEN (UNIT=3,NAME='AAVARSG',STATUS='OLD')
18.           READ(3,*) (VAR(I),I=1,600)
19.        CLOSE (UNIT=3,DISPOSE='KEEP')
20. C
21.                     OUTPUT(1)='AMMAE1'
22.                     OUTPUT(2)='AMMAE2'
23. C
24.        KK=0
25.        NNFILE=NFILE/2
26.        M=2*L+1
27. C
28.        DO 50 COUNT=1,2
29.           OPEN (UNIT=3,NAME=OUTPUT(COUNT),BLOCKSIZE=300,STATUS='NEW')
30. C
31.           DO 10 K=1,NNFILE
32.              KK=KK+1
33.              OPEN (UNIT=4,NAME=FNAME(KK),STATUS='OLD')
34.                 READ(4,*) (EMG(I),I=1,LAST)
35.              CLOSE(UNIT=4,DISPOSE='KEEP')
36. C
37.              DO 20 K2=1,3
38.                 I1=(KK-1)*3+K2
39.                 I=START(I1)/2-L-1
40. *               I=START(I1)-L-1
41. C
42.                 IF (VAR(I1) .LE. RATIO) THEN
43.                     NUM=NUM+1
44.                     DUM=0.0
45.                    DO 30 K3=1,M
46.                       DUM1=EMG(I+K3)**2
47.                       DUM=DUM+DUM1/M
48. 30                 CONTINUE
49.                    ME(1)=DUM
50.                    AVE(1)=AVE(1)+DUM/600.0
51. C
52.                       II=I
53.                    DO 40 K4=2,LEN
54.                       I=II+K4
55.                       DUM1=EMG(I+M-1)**2
56.                       DUM2=EMG(I-1)**2
57.                       DUM=DUM+(DUM1-DUM2)/M
58.                       ME(K4)=DUM
59.                       AVE(K4)=AVE(K4)+DUM/600.0
60. 40                 CONTINUE
61.                    CALL PLOT(ME,1,LEN,1)
62.                 END IF
63. 20           CONTINUE
64. 10        CONTINUE
```

```
65.           CLOSE (UNIT=3,DISPOSE='KEEP')
66. 50     CONTINUE
67. C
68.        OPEN (UNIT=3,NAME='AME',STATUS='NEW')
69.             WRITE(3,*) (AVE(I),I=1,LEN),NUM
70.             CALL PLOT(AVE,1,LEN,1)
71.        CLOSE(UNIT=3,DISPOSE='KEEP')
72.        STOP
73.        END
```

B.1.11 Program EMMAE.FTN

```
      PROGRAM EMMAE
      PARAMETER (LEN=90,BEGIN=30,IEND=70)
      INTEGER I,K,K1,COUNT1,NN
      REAL*4 AVE(LEN),ME(LEN),DUM,ERR,SUM,NORM,MSE
      CHARACTER*9 INPUT(4)
      DATA NORM/0.0/
C
      OPEN (UNIT=4,NAME='AAMEMG',STATUS='OLD')
           READ(4,*) (AVE(I),I=25,LEN)
      CLOSE (UNIT=4,DISPOSE='KEEP')
C
      DO 40 K=BEGIN,IEND
         DUM=AVE(K)
         NORM=NORM+DUM*DUM
40    CONTINUE
C
                        INPUT(1)='AMMAE1'
                        INPUT(2)='AMMAE2'
                        INPUT(3)='AMMAE3'
                        INPUT(4)='AMMAE4'
C
      OPEN (UNIT=4,NAME='EMMAE',STATUS='NEW')
C
      DO 10 COUNT1=1,4
         OPEN (UNIT=3,NAME=INPUT(COUNT1),STATUS='OLD')
              READ(3,*) NN
C              NN=2
           DO 20 K=1,NN
              READ(3,*) (ME(I),I=1,LEN)
              SUM=0.0
              DO 30 K1=BEGIN,IEND
                 ERR=AVE(K1)-ME(K1)
                 SUM=SUM+ERR*ERR
30            CONTINUE
              MSE=SUM/NORM*100.0
              WRITE(4,*) MSE
20         CONTINUE
         CLOSE (UNIT=3,DISPOSE='KEEP')
10    CONTINUE
      CLOSE (UNIT=4,DISPOSE='KEEP')
      STOP
      END
```

B.1.12 Program WIENER.FTN

```
 1.        PARAMETER (NN=50,IEND=40,JUMP=90,NFILE=50)
 2.        INTEGER*2 K,I,II,T,TT,N,START(150),NSTART,FNUM
 3.        REAL*4 K1,K2,KK,L,A,K4,K3,W5,XN(IEND),H(NN),X(5000),
 4.       C         WMM,WNN,WN,WM,DUM,SUM,DUM1,DUM2
 5.        CHARACTER*9 INPUT(50)
 6.        DATA XN/IEND*0.0/,SUM/0.0/,DUM2/0.0/
 7. C
 8.        WN=5
 9.        WM=100
10.        K1=1.0
11.        K2=1.0
12. C
13.        WNN=2.0*3.1416*WN
14.        WMM=2.0*3.1416*WM
15. C
16.        KK=K1+K2
17.        KK=SQRT(KK)
18.        L=(K1/(WMM*WMM)+K2/(WNN*WNN))
19.        L=SQRT(L)
20.        A=K1*(1.0+WNN/WMM)/(KK+WNN*L)
21.        K3=A/(L*WMM)
22.        K4=A/L*(1.0-KK/(L*WMM))
23.        W5=KK/(L)
24. C
25.        DO 10 K=1,NN
26.           DUM=EXP(-W5*(K-1)*0.002)*K4
27.           H(K)=DUM
28.           SUM=SUM+DUM
29. 10     CONTINUE
30.        H(1)=H(1)+K3
31.        SUM=SUM+K3
32. C
33.                INPUT(1)='AAS1'
34.               INPUT(50)='AAS50'
35. *
36.        OPEN (UNIT=3,NAME='AASTARTER',STATUS='OLD')
37.             READ(3,*) (START(I),I=1,NFILE*3)
38.        CLOSE (UNIT=3,DISPOSE='KEEP')
39. *
40.        OPEN (UNIT=3,NAME='AAW1',STATUS='NEW')
41. *
42.        DO 100 FNUM=1,NFILE
43.           OPEN (UNIT=4,NAME=INPUT(FNUM),STATUS='OLD')
44.                READ(4,*) (X(I),I=1,5000)
45.           CLOSE(UNIT=4,DISPOSE='KEEP')
46. *
47.           DO 110 COUNT=1,3
48.              DUM2=0.0
49.              II=START((FNUM-1)*3+COUNT)+JUMP
50. *
51. C
52.              DO 11 K=II-50,II-30
53.                 DUM2=DUM2+X(K)
54. 11           CONTINUE
55.              DUM2=DUM2/21.0
56. C
57.              DO 40 K=II-60,II+IEND
58. *               DUM1=ABS(X(K))
59.                 DUM1=X(K)-DUM2
60.                 DUM=DUM1*DUM1
61.                 DUM=LOG(DUM)
62.                 X(K)=DUM
63. 40           CONTINUE
64. C
```

```
65.                DO 41 K=1,IEND
66.  41               XN(K)=0.0
67.  C
68.                DO 20 T=II,II+IEND-1
69.                   TT=T-II+1
70.                   DO 30 K=1,NN
71.  30                  XN(TT)=XN(TT)+H(K)*X(T-K+1)
72.  20            CONTINUE
73.  C
74.                DO 200 K=II,II+IEND-1
75.                   TT=K-II+1
76.                   DUM=XN(TT)/SUM
77.                   DUM=EXP(DUM)
78.                   XN(TT)=DUM
79.  200           CONTINUE
80.  C
81.  C             WRITE(3,101) K1,K2,WN,WM
82.                WRITE(3,*) (XN(I),I=1,IEND)
83.  C             CALL PLOT(XN,1,IEND,1)
84.  c
85.  110        CONTINUE
86.  100     CONTINUE
87.  c
88.         CLOSE (UNIT=3,DISPOSE='KEEP')
89.  C
90.         STOP
91.  101          FORMAT(10X,'K1=',F7.2,3X,'K2=',F7.2,3X,'WN=',F7.2,
92.       C               3X,'WM=',F7.2/)
93.         END
```

B.1.13 Program WIENEE.FTN

```
      PROGRAM WIENEE
      PARAMETER (LAST=40,JUMP=0,SCALE=1.0,NFILE=4,NSAM=150)
      INTEGER I,K,K1,COUNT1,NN,N(LAST),FNUM,COUNT
      REAL*4 AVE(LAST),ME(LAST),DUM,ERR,NORM,MSE,SUM
      CHARACTER*9 INPUT(4)
      DATA NORM/0.0/,COUNT/0/
C
      OPEN (UNIT=4,NAME='AAMEMG',STATUS='OLD')
           READ(4,*) (N(I),AVE(I),I=1,LAST)
      CLOSE (UNIT=4,DISPOSE='KEEP')
C
      DO 40 K=1,LAST
         DUM=AVE(K)
         NORM=NORM+DUM*DUM
40    CONTINUE
C
               INPUT(1)='AAW1'
               INPUT(2)='AAW2'
               INPUT(3)='AAW3'
               INPUT(4)='AAW4'
C
      OPEN (UNIT=4,NAME='AAWE',STATUS='NEW')
        WRITE(4,*) SCALE
C
          DO 100 FNUM=1,NFILE
C
             OPEN (UNIT=3,NAME=INPUT(FNUM),STATUS='OLD')
               DO 20 K=1,NSAM
                  READ(3,*) (ME(I),I=1,LAST)
                  SUM=0.0
                  DO 30 K1=1,LAST
                     ERR=AVE(K1)-ME(K1)*SCALE
                     SUM=SUM+ERR*ERR
30                CONTINUE
                  MSE=SUM/NORM*100.0
                  IF (MSE .LE. 30.00) THEN
                     COUNT=COUNT+1
                  END IF
                  WRITE(4,*) K,MSE
20             CONTINUE
             CLOSE (UNIT=3,DISPOSE='KEEP')
100       CONTINUE
      WRITE(4,*) COUNT
      CLOSE (UNIT=4,DISPOSE='KEEP')
      STOP
      END
```

B.2 CURVE-FITT PROGRAMS USED IN THE AMDAHL

B.2.1 Program CFIT

```
1. //  JOB ',,,L=5,T=20,I=8',CLASS=1
2. // EXEC SASPLOT,OPTIONS='S=80'
3. //SYSIN DD *
4. GOPTIONS DEVICE=XEROX HSIZE=10.75 VSIZE=8.25 COLORS=(RED,BLUE)
5.                                    ROTATE;
6.        DATA RMS;
7.         INPUT T FIT@@;
8.          K=0.31488;
9.         OUTPUT;
10.        CARDS;
```

```
200.        ;
201.       PROC NLIN
202.        DATA=RMS
203.        METHOD=DUD;
204.       PARMS
205.        K1=0.01 TO 1.0 BY 0.1;
206.       DUM=EXP(-K1*T);
207.       MODEL FIT=K*(1-DUM);
208.       OUTPUT OUT=B R=RMSERR P=PREDICT;
209.       PROC GPLOT DATA=B;
210.        PLOT PREDICT*T FIT*T/OVERLAY;
211.        TITLE .C=RED .F=TRIPLEX .H=2   VEN-EMG(C) 65;
212.        SYMBOL1 V=PLUS C=BLUE I=SPLINE;
213.        SYMBOL2 V=+ C=RED I=SPLINE;
214.        FOOTNOTE1 .C=RED .F=DUPLEX .H=0.7 PLUS SIGN = PREDICTED CURVE;
215.        FOOTNOTE2 .C=RED .F=DUPLEX .H=0.7 CIRCLED PLUS = ACTUAL CURVE;
216.        FOOTNOTE3 .C=BLUE .F=DUPLEX .H=1 T= *2mS;
217. /*
218. //S2 EXEC XPLOT
219. //
```

B.2.2 Program DFIT

```
1. //  JOB ',,,L=5,T=4M,I=20',CLASS=1
2. // EXEC SASPLOT,OPTIONS='S=80'
3. //SYSIN DD *
4. GOPTIONS DEVICE=XEROX HSIZE=10.75 VSIZE=8.25 COLORS=(RED,BLUE)-
5.                                      ROTATE;
6.       DATA RMS;
7.        INPUT T FIT@@;
8.           K=0.10967;
9.          OUTPUT;
10.         CARDS;
```

```
210.          ;
220.       PROC NLIN
230.         BEST=5
240.         DATA=RMS
250.         METHOD=DUD;
260.       PARMS
270.         A=-1 TO 1 BY 0.1
280.         B=-1 TO 1 BY 0.1;
290.       DUM1=EXP(-A*T);
300.       DUM2=EXP(-B*T);
310.       DUM3=A-B;
320.       K1=B/DUM3;
330.       K2=-A/DUM3;
340.       MODEL FIT=K*(1+K1*DUM1+K2*DUM2);
350.       DER.A=-A*K1*DUM1;
360.       DER.B=-B*K2*DUM2;
370.       OUTPUT OUT=B R=RMSERR P=PREDICT;
380.       PROC GPLOT DATA=B;
390.        PLOT PREDICT*T FIT*T/OVERLAY;
400.        TITLE .C=RED .F=TRIPLEX .H=2  FKEN-EMG(D) 11;
410.        SYMBOL1 V=PLUS C=BLUE I=SPLINE;
420.        SYMBOL2 V=+ C=RED I=SPLINE;
430.        FOOTNOTE1 .C=RED .F=DUPLEX .H=1 PLUS SIGN = PREDICTED CURVE;
440.        FOOTNOTE2 .C=RED .F=DUPLEX .H=1 CIRCLED PLUS = ACTUAL CURVE;
450.        FOOTNOTE3 .C=BLUE .F=TRIPLEX .H=1 T= *2mS;
460. /*
470. //S2 EXEC XPLOT
480. //
```

B.2.3 Program EDPARK

```
 6. //  JOB ',,,L=5,T=20,I=8',CLASS=A
 7. // EXEC SASPLOT,OPTIONS='S=80'
 8. //SYSIN DD *
 9. GOPTIONS DEVICE=XEROX HSIZE=10.75 VSIZE=8.25 COLORS=(RED,BLUE)
10.                                     ROTATE;
11.       DATA RMS;
12.        INPUT TAU FIT@@;
13.         E3=0.05083;
14.         ACM=1;
15.         OFFSET=0.07210;
16.         K=0.43734;
17.         ZERO=47;
18.         T=60;
19.        OUTPUT;
20.        CARDS;
600.          ;
601.        PROC NLIN
602.          DATA=RMS
603.          METHOD=DUD;
604.        PARMS
605.         A=0.1 TO 2.0 BY 0.1;
606.        TAU1=TAU-ZERO;
607.        TT1=T-ZERO;
608.        NT=(K*(1-EXP(-E3*TT1))+OFFSET);
609.        NTAU=(K*(1-EXP(-E3*TAU1))+OFFSET);
610.        T1=TAU1-TT1;
611.        K1=ACM*(A**3);
612.        DUM1=(1/(A**3)+ABS(T1)/(A*A)-T1*T1/A)*EXP(-A*ABS(T1));
613.        MODEL FIT=K1*DUM1*SQRT(NT)*SQRT(NTAU);
614.        OUTPUT OUT=B R=RMSERR P=PREDICT;
615.        PROC GPLOT DATA=B;
616.          PLOT PREDICT*TAU FIT*TAU/OVERLAY;
617.          TITLE1 .C=RED .F=TRIPLEX .H=2 KKEN(1P) ZERO=47;
618.          TITLE2 .C=RED .F=DUPLEX .H=2 T=60;
619.          LABEL PREDICT='ATUO-CORRELATION';
620.          SYMBOL1 V=PLUS C=BLUE I=SPLINE;
621.          SYMBOL2 V=+ C=RED I=SPLINE;
622.          FOOTNOTE .C=BLUE .F=DUPLEX .H=1.3 T=*2mS;
623. /*
624. //S2 EXEC XPLOT
625. //
  5. //  JOB ',,,L=5,T=20,I=8',CLASS=A
  6. // EXEC SASPLOT,OPTIONS='S=80'
```

B.2.4 Program PARK2

```
1. //  JOB ',,,L=5,T=20,I=8',CLASS=1
2. // EXEC SASPLOT,OPTIONS='S=80'
7. //SYSIN DD *
8. GOPTIONS DEVICE=XEROX HSIZE=10.75 VSIZE=8.25 COLORS=(RED,BLUE)
9.                                    ROTATE;
10.       DATA RMS;
11.        INPUT TAU FIT@@;
12.         A1=0.20175;
13.         B1=0.20434;
14.         ACM=1;
15.         OFFSET=0.00823;
16.         K=0.11377;
17.         ZERO=56;
18.         T=154;
19.        OUTPUT;
20.        CARDS;
```

```
550.        ;
560.       PROC NLIN
570.        DATA=RMS
580.        METHOD=DUD;
590.       PARMS
600.        A=0.1 TO 2.0 BY 0.1;
610.       TAU1=TAU-ZERO;
620.       TT1=T-ZERO;
630.       DUM11=EXP(-A1*TAU1);
640.       DUM12=EXP(-B1*TAU1);
650.       DUM21=EXP(-A1*TT1);
660.       DUM22=EXP(-B1*TT1);
670.       DUM3=A1-B1;
680.       KK1=B1/DUM3;
690.       KK2=-A1/DUM3;
700.       NT=(K*(1+KK1*DUM11+KK2*DUM12)+OFFSET);
710.       NTAU=(K*(1+KK1*DUM21+KK2*DUM22)+OFFSET);
720.       T1=TAU1-TT1;
730.       K1=ACM*(A**3);
740.       MUM1=(1/(A**3)+ABS(T1)/(A*A)-T1*T1/A)*EXP(-A*ABS(T1));
750.       MODEL FIT=K1*MUM1*SQRT(NT)*SQRT(NTAU);
760.       OUTPUT OUT=B R=RMSERR P=PREDICT;
770.       PROC GPLOT DATA=B;
780.        PLOT PREDICT*TAU FIT*TAU/OVERLAY;
790.        TITLE .C=RED .F=TRIPLEX .H=2 ANG(2P) ZERO=56;
800.        TITLE2 .C=RED .F=TRIPLEX .H=2 T=154;
810.        SYMBOL1 V=PLUS C=BLUE I=SPLINE;
820.        SYMBOL2 V=+ C=RED I=SPLINE;
830.        LABEL PREDICT='AUTO-CORRELATION';
840.        FOOTNOTE .C=RED .F=DUPLEX .H=1 T=*2mS;
850. /*
860. //S2 EXEC XPLOT
870. //
```

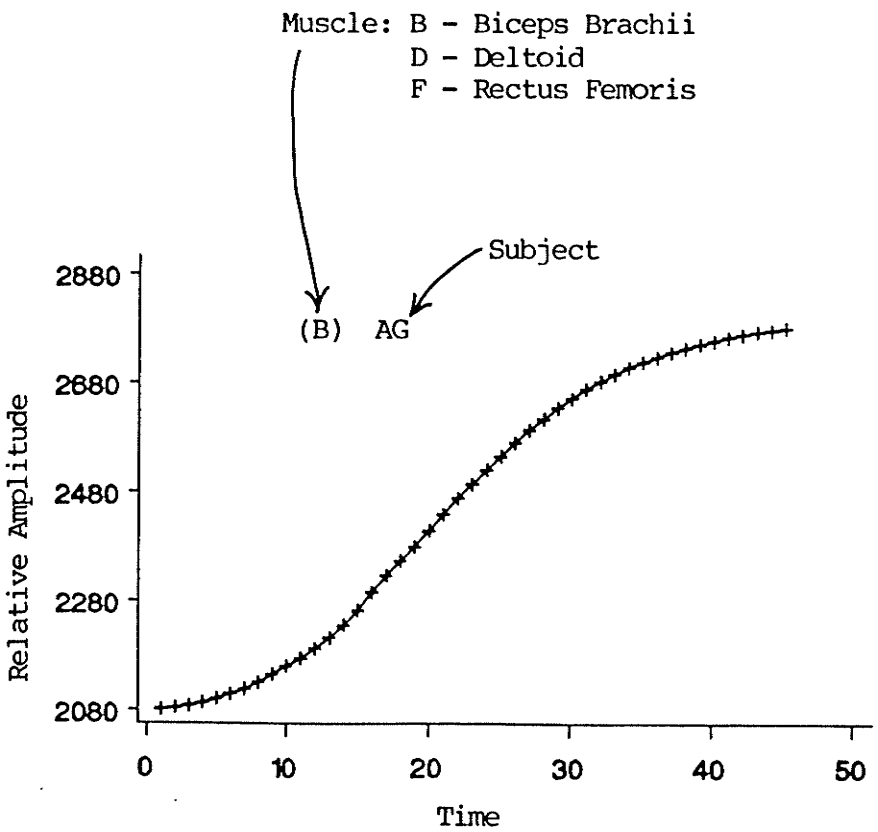
B.2.5 Subroutine PLOT

```
 1.        SUBROUTINE PLOT(F,M,N,L)
 2.        DIMENSION F(N)
 3.        REAL LINE(65)
 4.        DATA BLANK,DOT,X/1H ,1H.,1H*/
 5.        A=0.
 6.        DO 20 K=M,N
 7.           B=ABS(F(K))
 8.           IF(B.GT.A) THEN
 9.              A=B
10.              P1=K
11.           END IF
12.  20    CONTINUE
13.        A=0.
14.        DO 30 K=M,N,L
15.           B=ABS(F(K))
16.           IF(B .GT. A) THEN
17.              A=B
18.              P2=K
19.           END IF
20.  30    CONTINUE
21.        IF (F(P1) .GT. F(P2)) THEN
22.            A=F(P1)
23.            F(P2)=A
24.        END IF
25.        A=A/32
26.        DO 10 J=1,65
27.  10       LINE(J)=BLANK
28.        LINE(33)=DOT
29.        DO 40 J=M,N,L
30.           K=INT(F(J)/A+33)
31.           LINE(K)=X
32.           WRITE(3,100)J,F(J),LINE
33.  100      FORMAT(1X,I4,F9.5,1X,65A1)
34.           LINE(K)=BLANK
35.  40       LINE(33)=DOT
36.        RETURN
37.        END
```

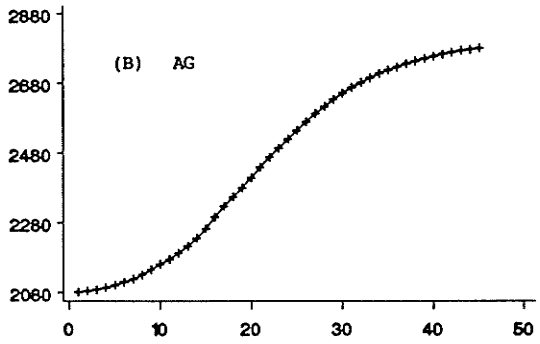
Appendix C

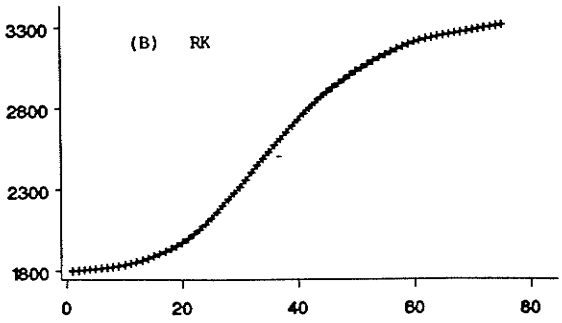
COMPUTATION RESULTS

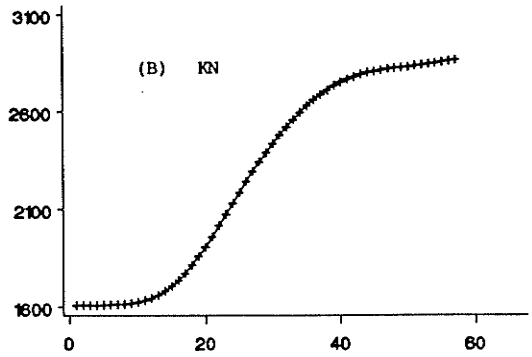
C.1 ENSEMBLE MEAN OF EXPERIMENTAL STRAIN-GAUGE DATA

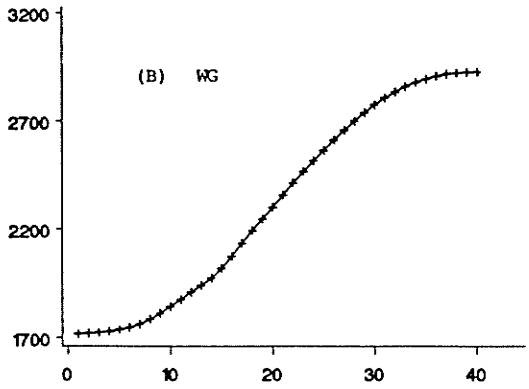


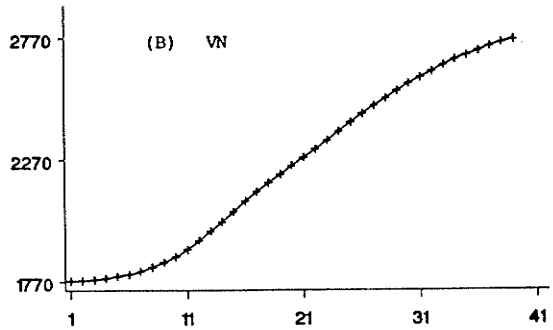
(to obtain time in mS multiply by 2.)

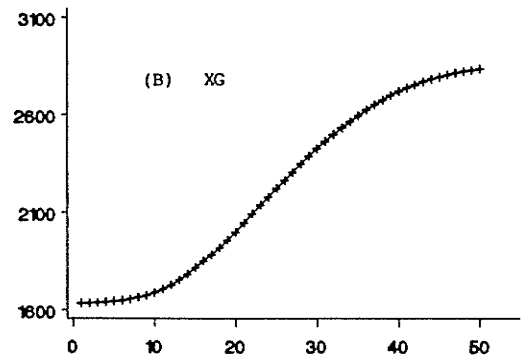

(B) AG
2880
2680
2480
2280
2080
0
10
20
30
40
50

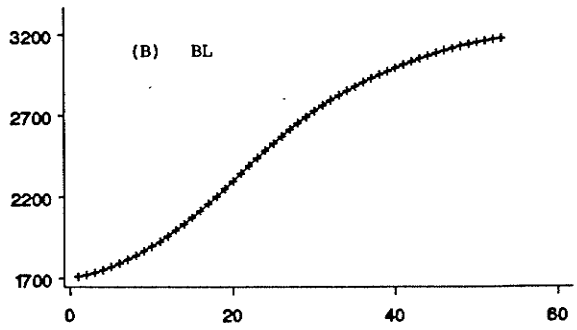

(B) RK
3300
2800
2300
1800
0
20
40
60
80

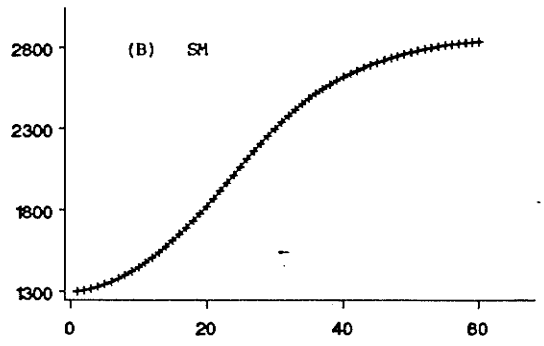

(B) KN
3100
2600
2100
1600
0
20
40
60

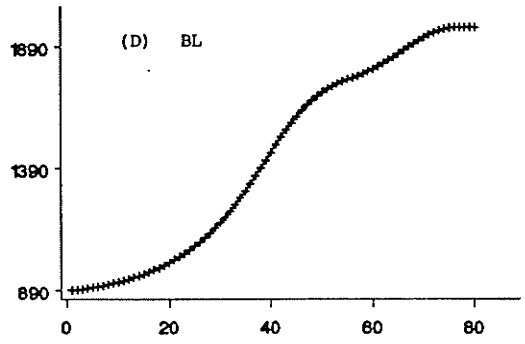

(B) WG
3200
2700
2200
1700
0
10
20
30
40

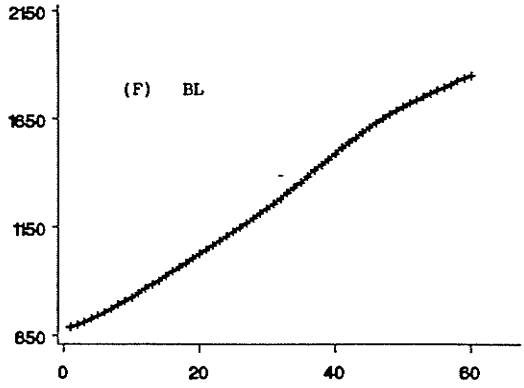

(B) VN
2770
2270
1770
1
11
21
31
41

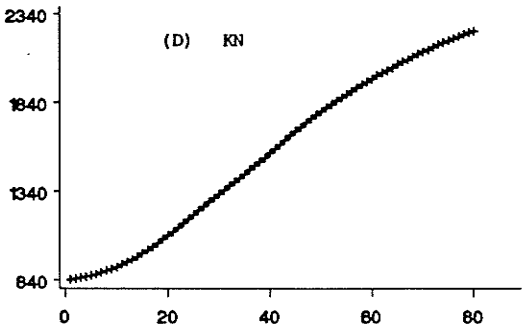

(B) XG
3100
2600
2100
1600
0
10
20
30
40
50

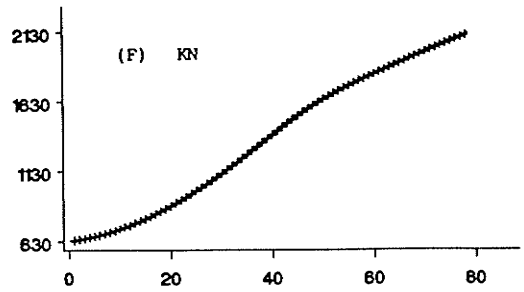

(B) BL
3200
2700
2200
1700
0
20
40
60

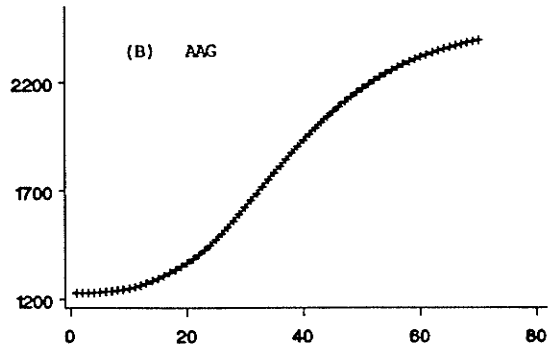

(B) SH
2800
2300
1800
1300
0
20
40
60

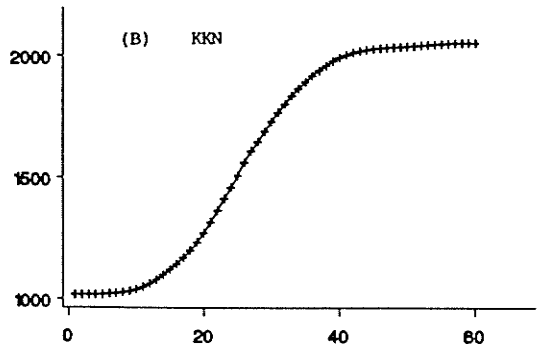

(D) BL
1890
1390
890
0
20
40
60
80

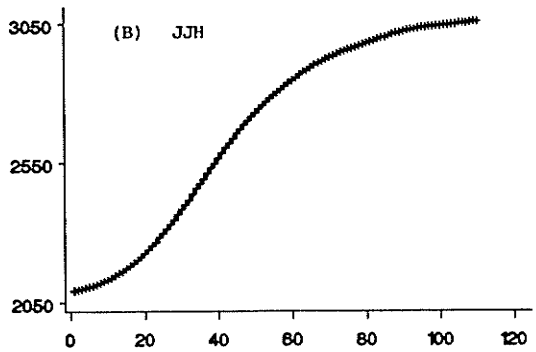

(F) BL
2150
1650
1150
650
0
20
40
60


(D) KN
2340
1840
1340
840
0
20
40
60
80

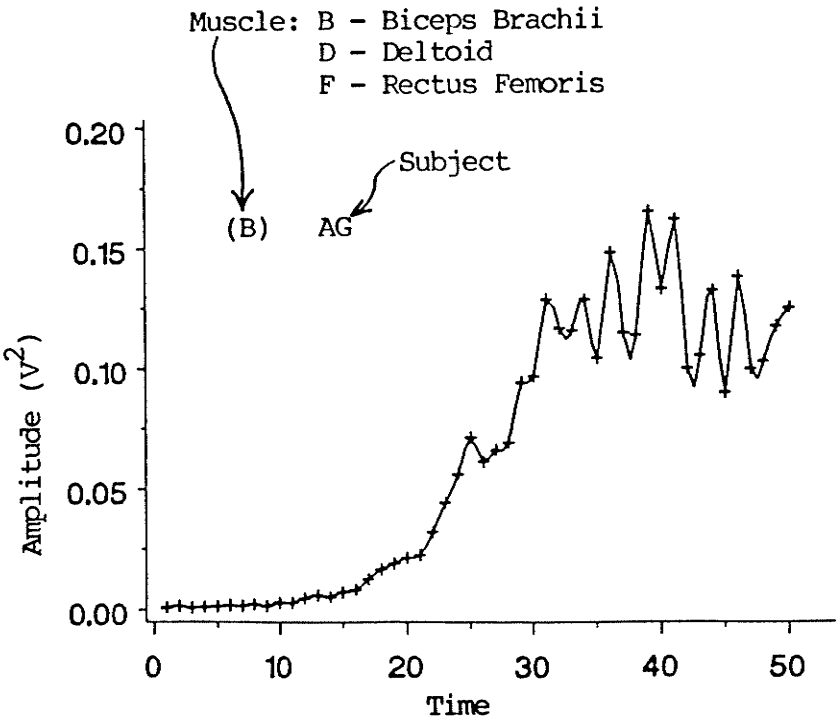

(F) KN
2130
1630
1130
630
0
20
40
60
80


(B) AAG
2200
1700
1200
0
20
40
60
80

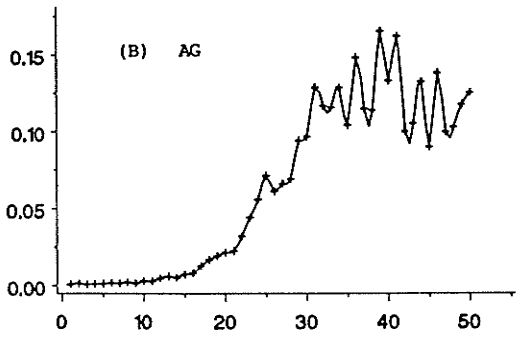

(B) KKN
2000
1500
1000
0
20
40
60

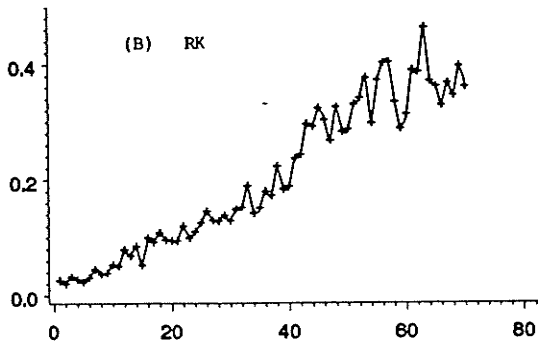

(B) JJH
3050
2550
2050
0
20
40
60
80
100
120

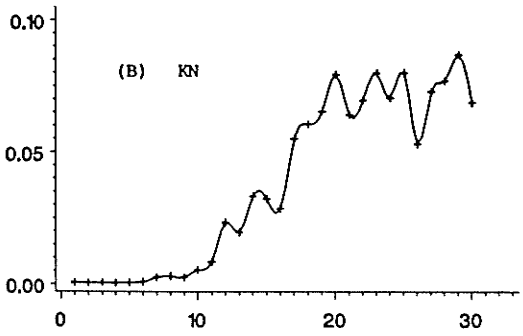
C.2 ENSEMBLE MEAN OF EXPERIMENTAL EMG VARIANCE

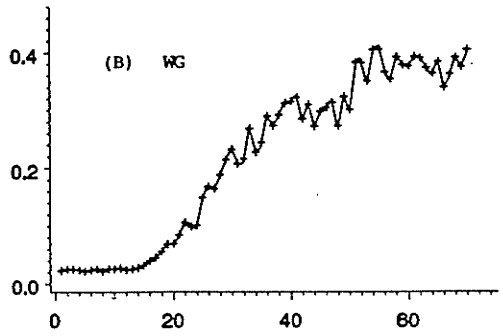


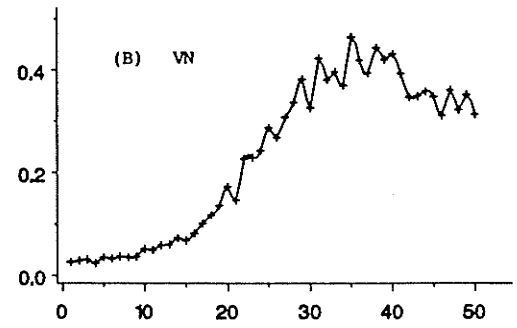
(to obtain time in mS multiply by 2.)

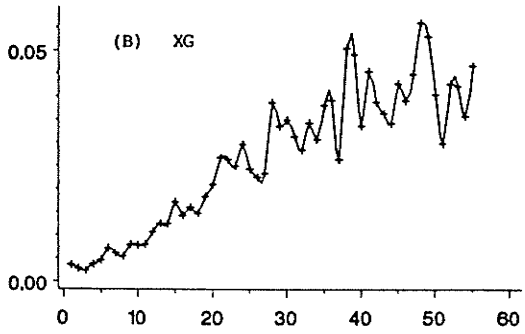

(B) AG
0.15
0.10
0.05
0.00
0
10
20
30
40
50

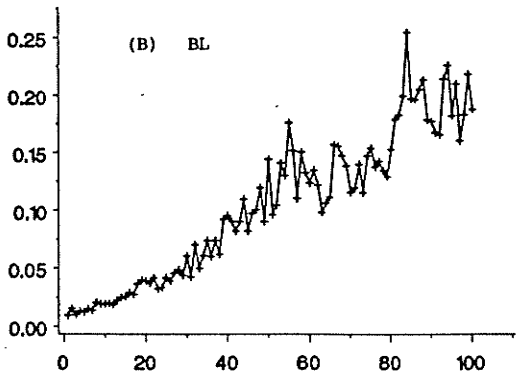

(B) RK
0.4
0.2
0.0
0
20
40
60
80

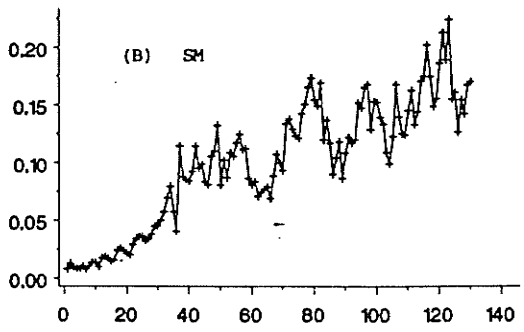

(B) KN
0.10
0.05
0.00
0
10
20
30

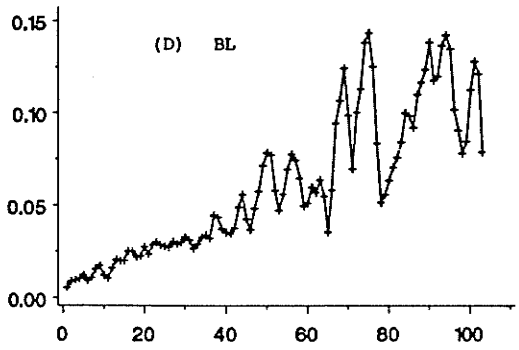

(B) WG
0.4
0.2
0.0
0
20
40
60

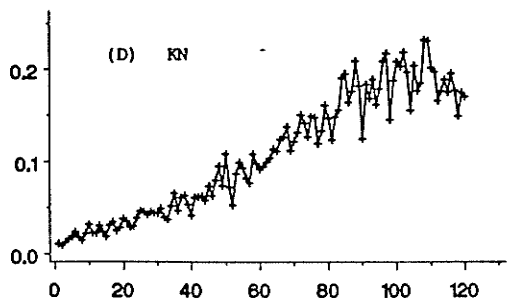

(B) VN
0.4
0.2
0.0
0
10
20
30
40
50

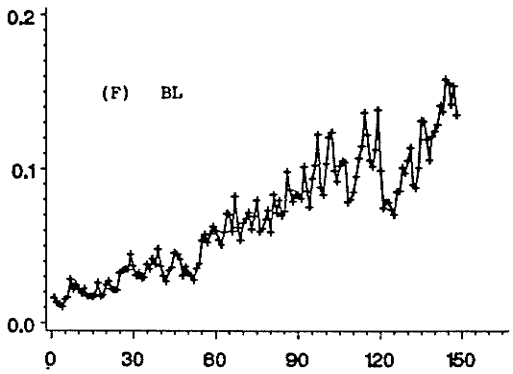

(B) XG
0.05
0.00
0
10
20
30
40
50
60

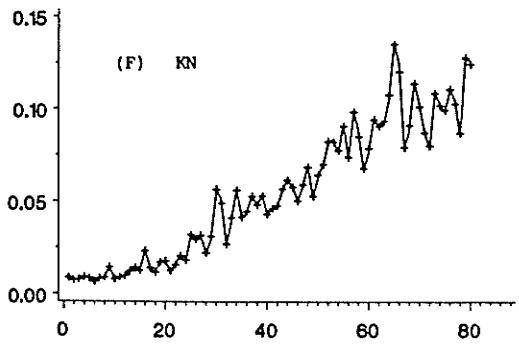

(B) BL
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
100

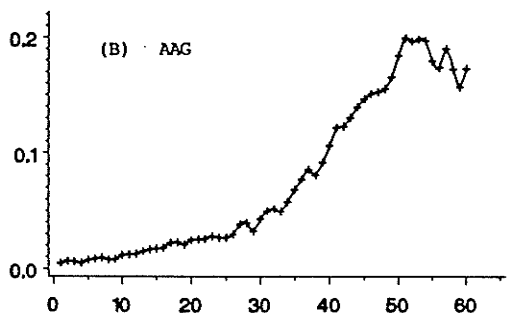

(B) SM
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
100
120
140

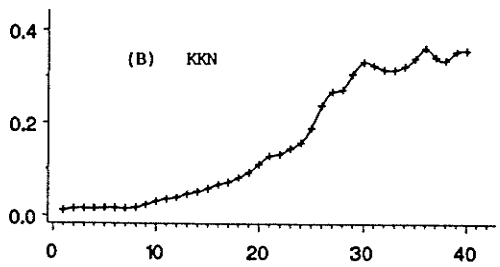

(D) BL
0.15
0.10
0.05
0.00
0
20
40
60
80
100

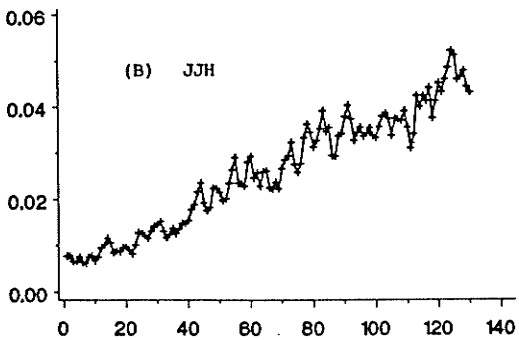

(D) KN
0.2
0.1
0.0
0
20
40
60
80
100
120


(F) BL
0.2
0.1
0.0
0
30
60
90
120
150


(F) KN
0.15
0.10
0.05
0.00
0
20
40
60
80

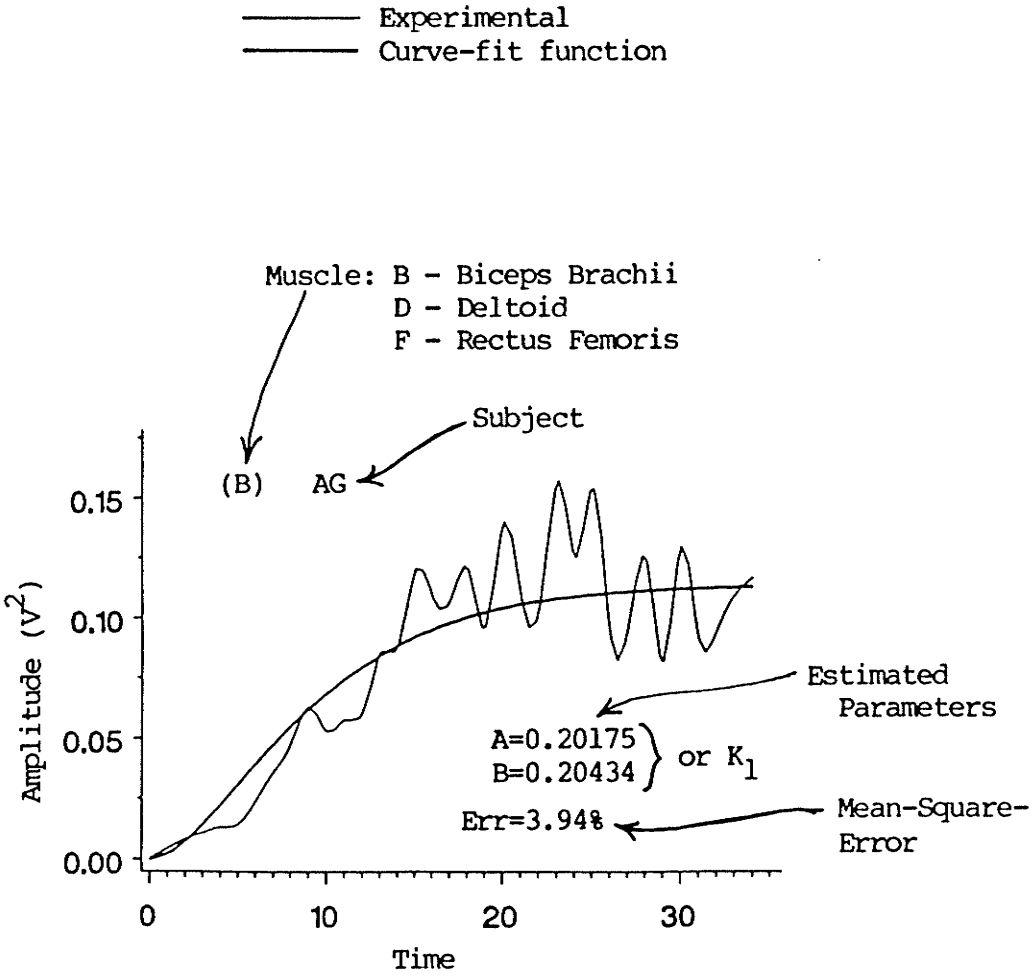

(B) AAG
0.2
0.1
0.0
0
10
20
30
40
50
60


(B) KKN
0.4
0.2
0.0
0
10
20
30
40

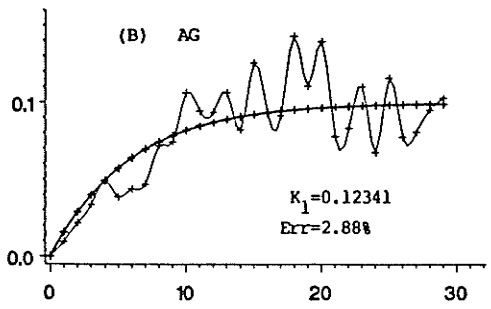

(B) JJH
0.06
0.04
0.02
0.00
0
20
40
60
80
100
120
140

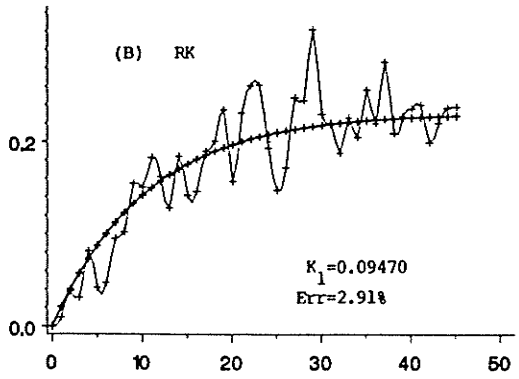
C.3 CURVE-FITTED EMG VARIANCE

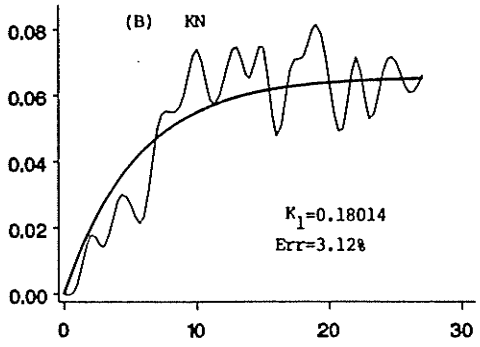
C.3.1 Curve-fitted Results with Function $\hat{\sigma}^2_{e_1}$

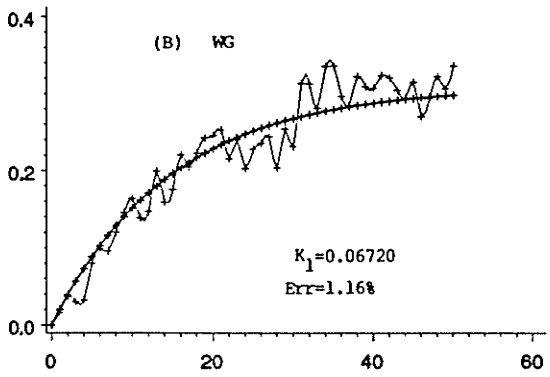


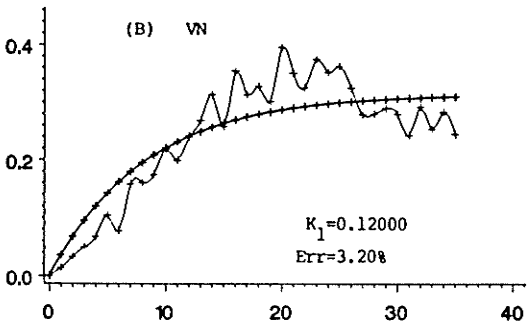
(to obtain time in mS multiply by 2.)

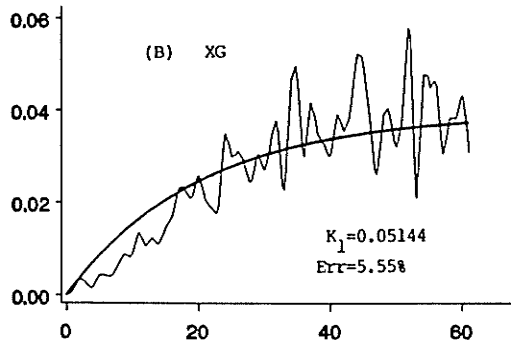

(B) AG
K_1=0.12341
Err=2.88%
0.0
0.1
0
10
20
30

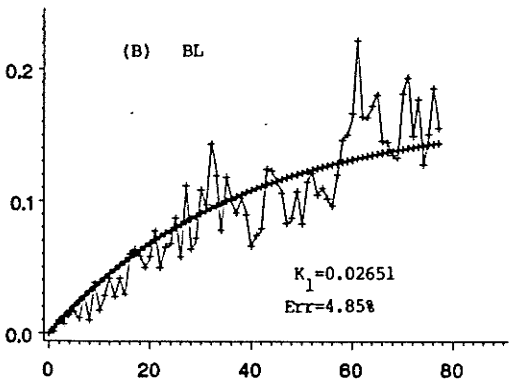

(B) RK
K_1=0.09470
Err=2.91%
0.0
0.2
0
10
20
30
40
50

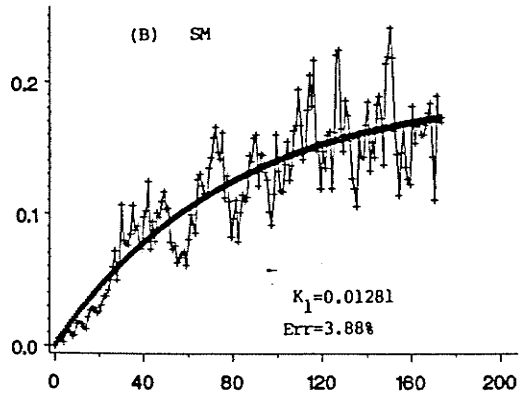

(B) KN
K_1=0.18014
Err=3.12%
0.00
0.02
0.04
0.06
0.08
0
10
20
30

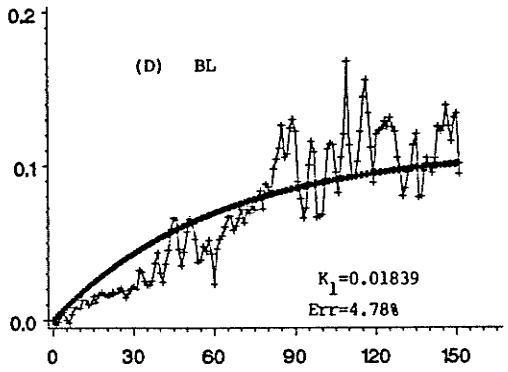

(B) WG
K_1=0.06720
Err=1.16%
0.0
0.2
0.4
0
20
40
60

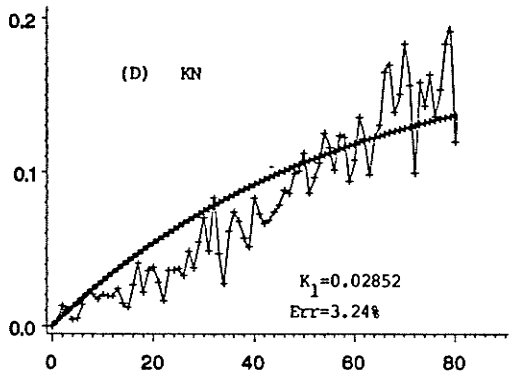

(B) VN
K_1=0.12000
Err=3.20%
0.0
0.2
0.4
0
10
20
30
40

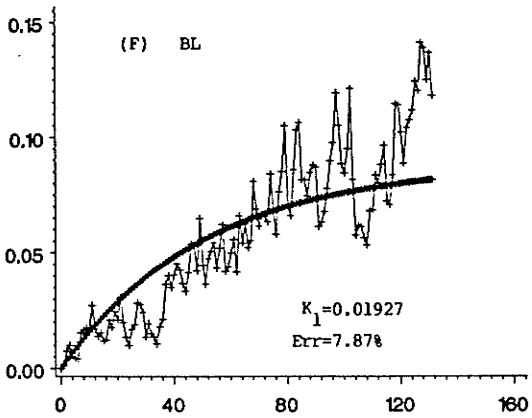

(B) XG
K_1=0.05144
Err=5.55%
0.00
0.02
0.04
0.06
0
20
40
60

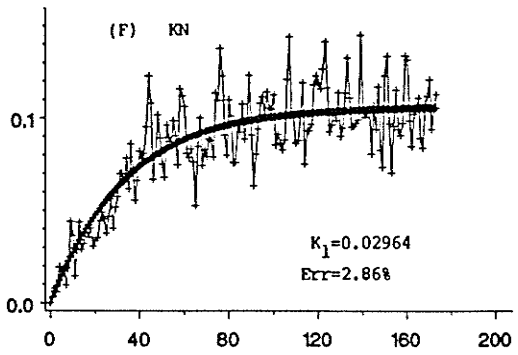

(B) BL
K_1=0.02651
Err=4.85%
0.0
0.1
0.2
0
20
40
60
80

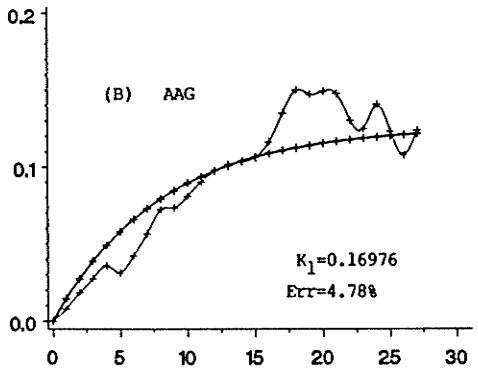

(B) SM
K_1=0.01281
Err=3.88%
0.0
0.1
0.2
0
40
80
120
160
200

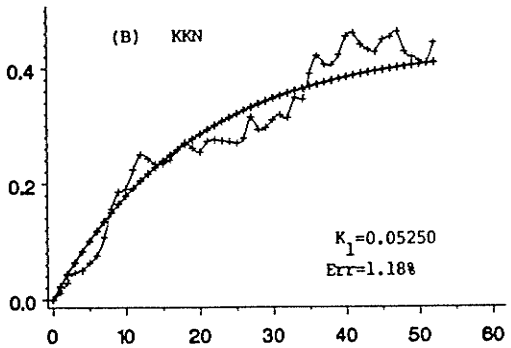

(D) BL
K_1=0.01839
Err=4.78%
0.2
0.1
0.0
0
30
60
90
120
150

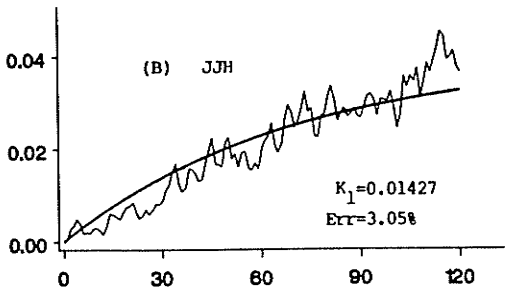

(D) KN
K_1=0.02852
Err=3.24%
0.2
0.1
0.0
0
20
40
60
80


(F) BL
K_1=0.01927
Err=7.87%
0.15
0.10
0.05
0.00
0
40
80
120
160

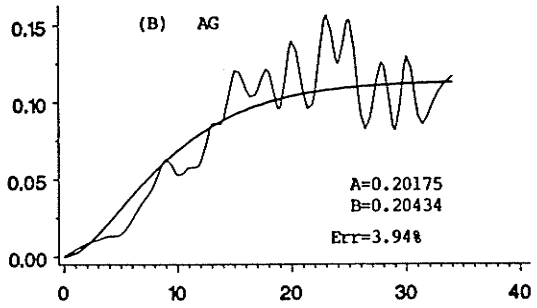

(F) KN
K_1=0.02964
Err=2.86%
0.1
0.0
0
40
80
120
160
200

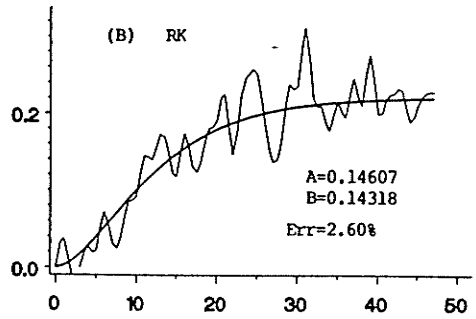

(B) AAG
K_1=0.16976
Err=4.78%
0.2
0.1
0.0
0
5
10
15
20
25
30

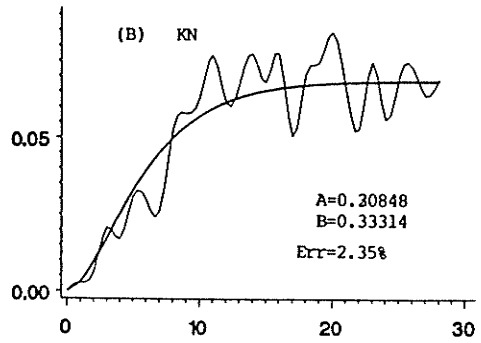

(B) KKN
K_1=0.05250
Err=1.18%
0.4
0.2
0.0
0
10
20
30
40
50
60

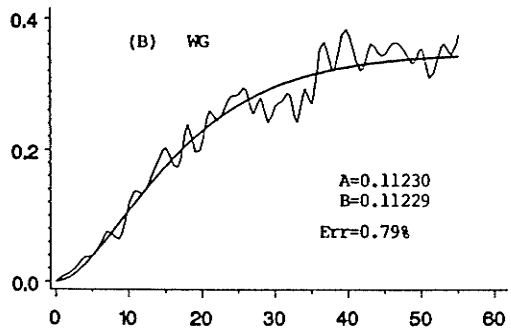

(B) JJH
K_1=0.01427
Err=3.05%
0.04
0.02
0.00
0
30
60
90
120

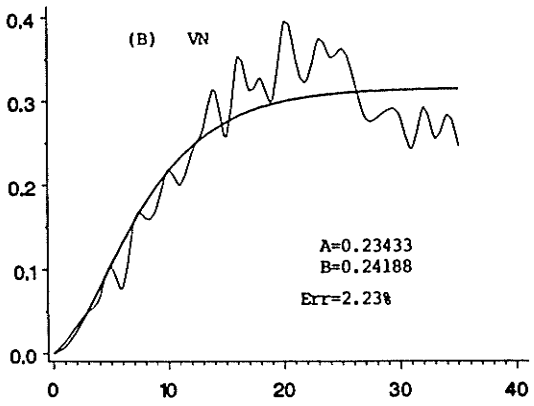
C.3.2 Curve-fitted Results with Function δe_2^2

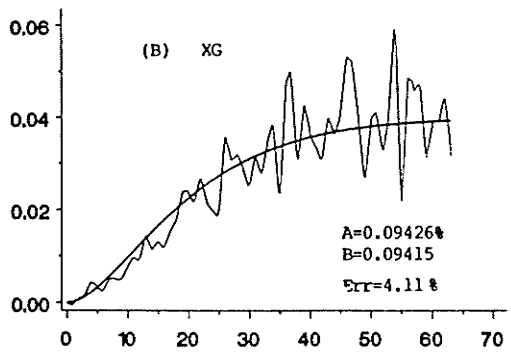


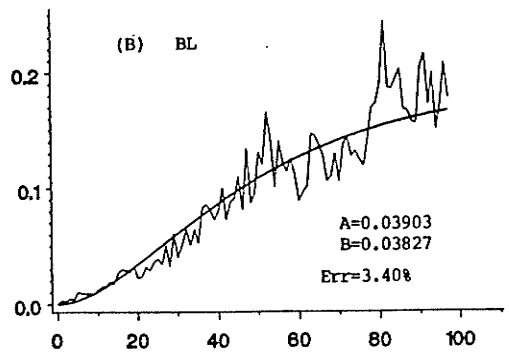


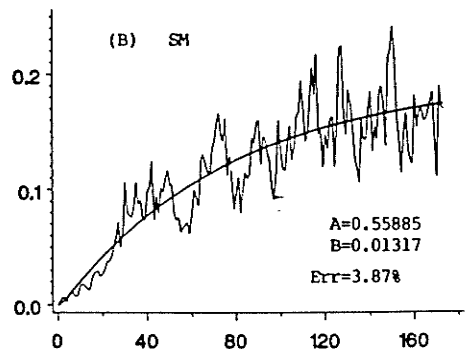


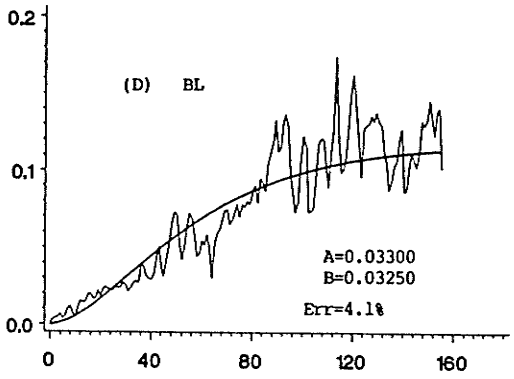


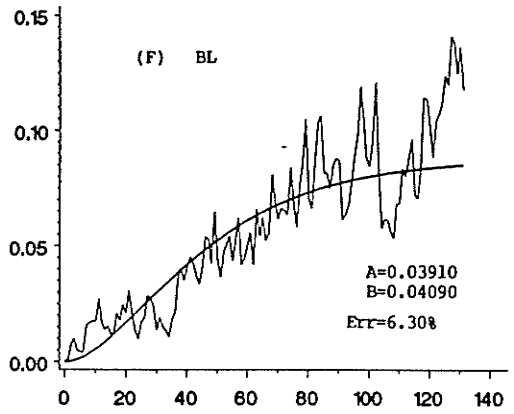


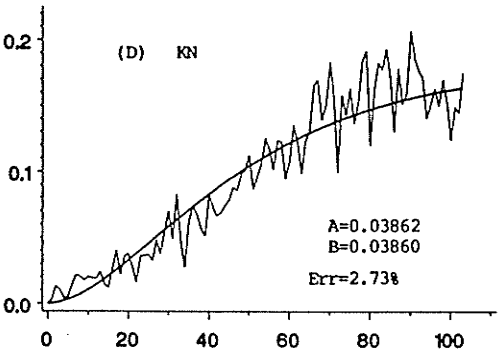


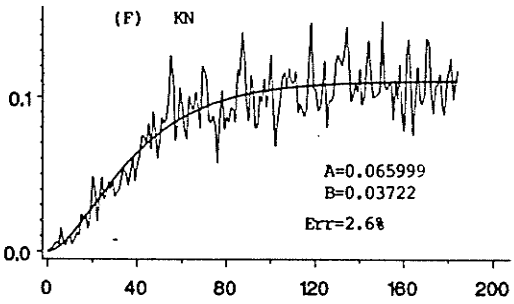


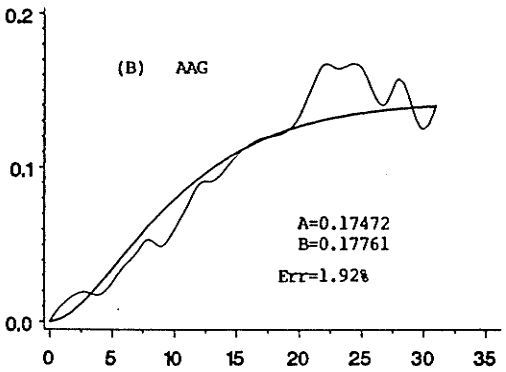


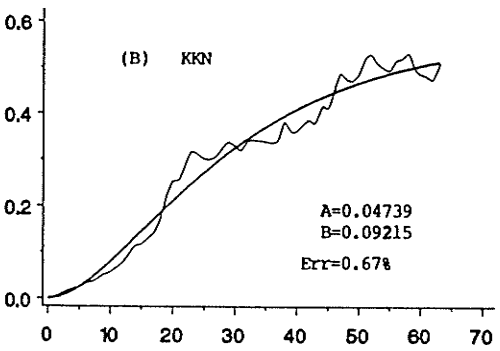

(D) BL
A=0.03300
B=0.03250
Err=4.1%
0.2
0.1
0.0
0
40
80
120
160

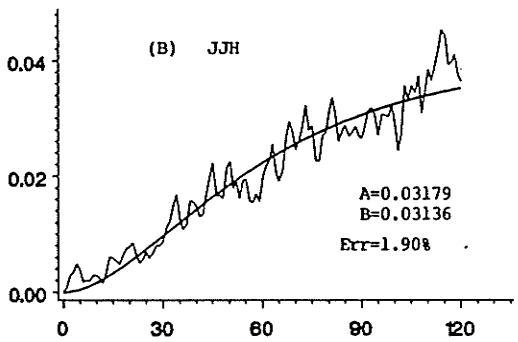

(F) BL
A=0.03910
B=0.04090
Err=6.30%
0.15
0.10
0.05
0.00
0
20
40
60
80
100
120
140


(D) KN
A=0.03862
B=0.03860
Err=2.73%
0.2
0.1
0.0
0
20
40
60
80
100


(F) KN
A=0.065999
B=0.03722
Err=2.6%
0.1
0.0
0
40
80
120
160
200

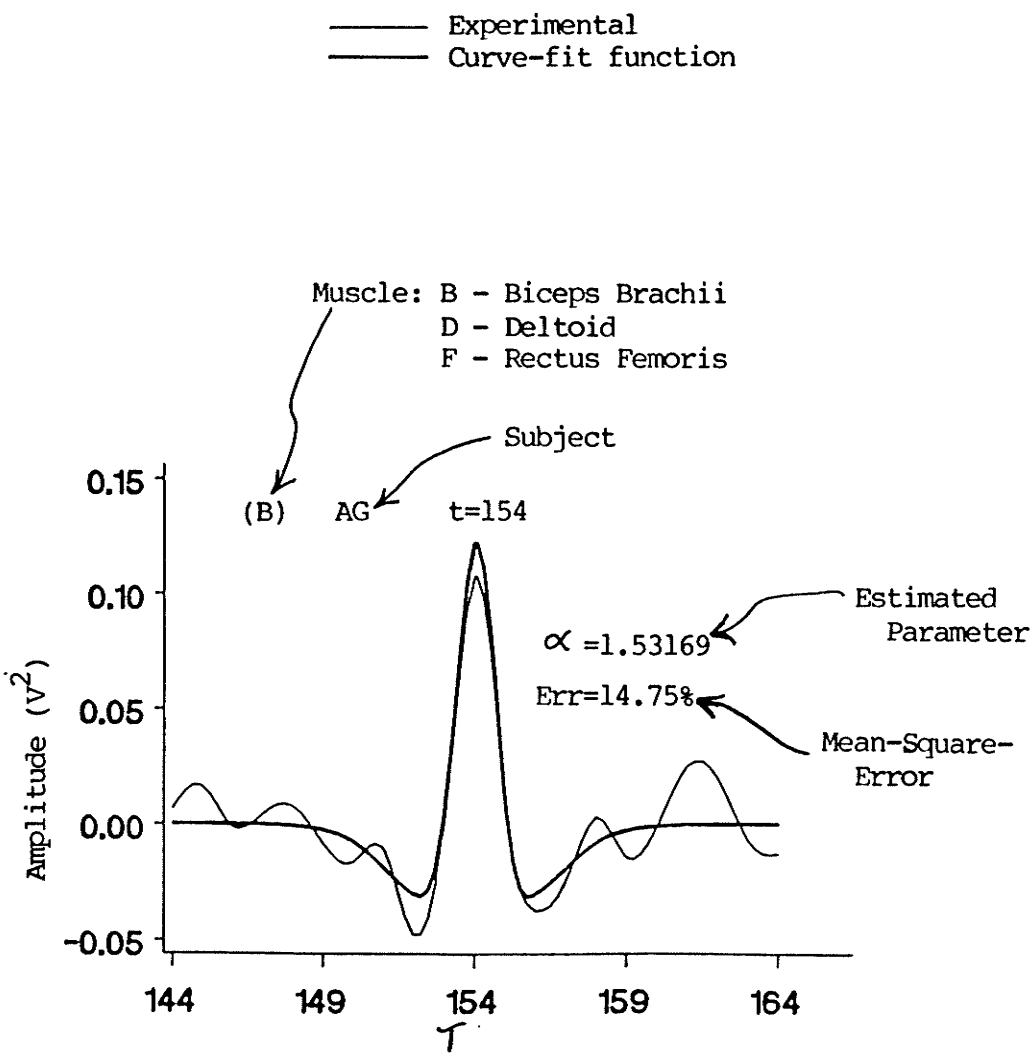

(B) AAG
A=0.17472
B=0.17761
Err=1.92%
0.2
0.1
0.0
0
5
10
15
20
25
30
35


(B) KKN
A=0.04739
B=0.09215
Err=0.67%
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
70


(B) JJH
A=0.03179
B=0.03136
Err=1.90%
0.04
0.02
0.00
0
30
60
90
120

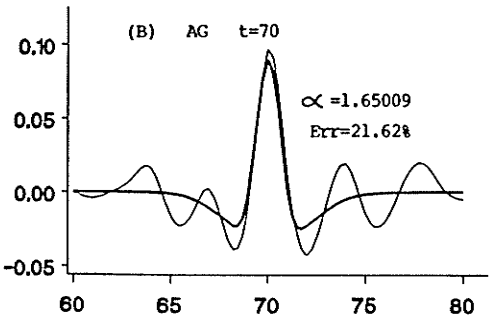
C.4 CURVE-FITTED AUTOCORRELATIONS WITH R_{ee}

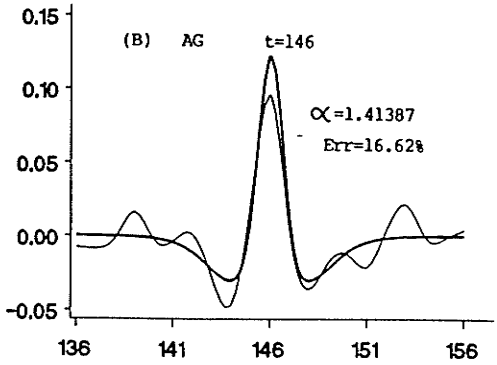
C.4.1 Combined with $\acute{o}e_1$

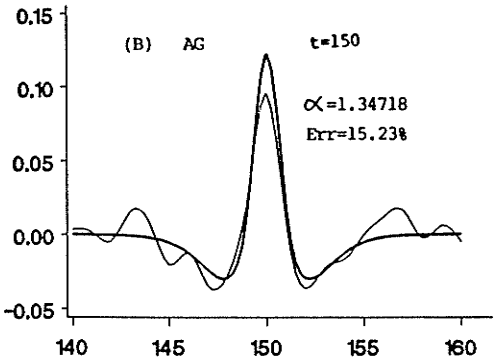


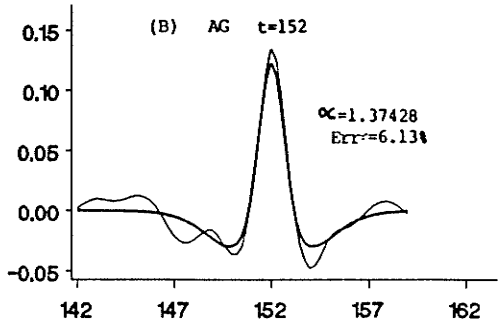
(to obtain τ in mS multiply by 2.)

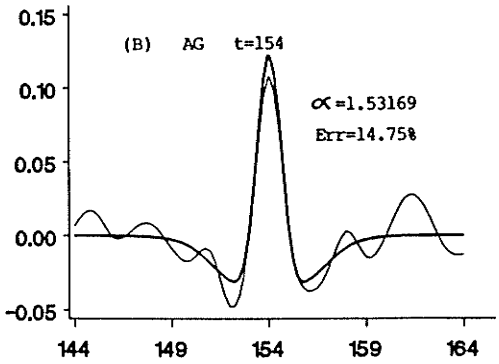
- where t is the time instant(from the defined starting-point) at which the autocorrelation is calculated.

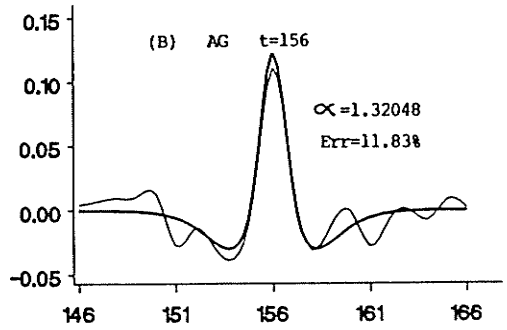

(B) AG t=70
∝ =1.65009
Err=21.62%
0.15
0.10
0.05
0.00
-0.05
60 65 70 75 80

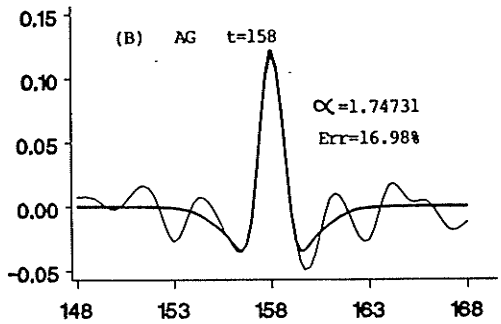

(B) AG t=146
∝=1.41387
Err=16.62%
0.15
0.10
0.05
0.00
-0.05
136 141 146 151 156

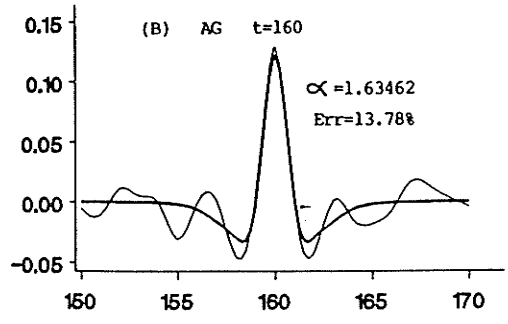

(B) AG t=150
∝=1.34718
Err=15.23%
0.15
0.10
0.05
0.00
-0.05
140 145 150 155 160

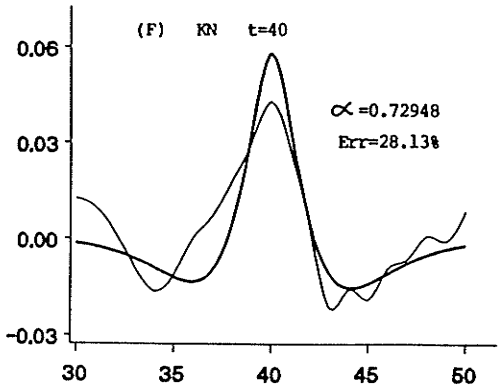

(B) AG t=152
∝=1.37428
Err=6.13%
0.15
0.10
0.05
0.00
-0.05
142 147 152 157 162

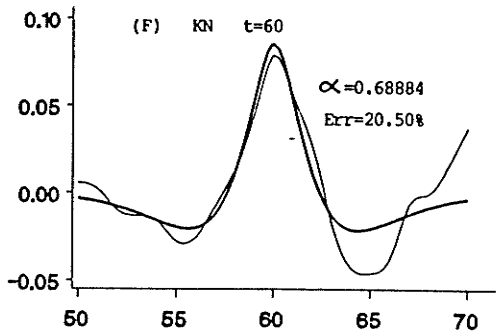

(B) AG t=154
∝=1.53169
Err=14.75%
0.15
0.10
0.05
0.00
-0.05
144 149 154 159 164

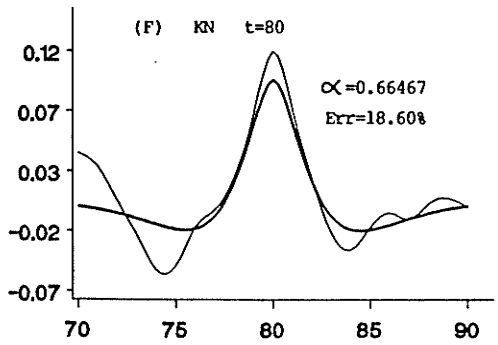

(B) AG t=156
∝=1.32048
Err=11.83%
0.15
0.10
0.05
0.00
-0.05
146 151 156 161 166

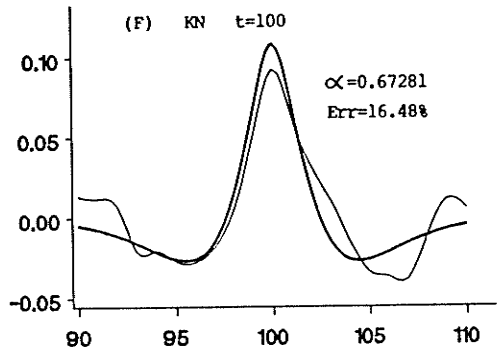

(B) AG t=158
∝=1.74731
Err=16.98%
0.15
0.10
0.05
0.00
-0.05
148 153 158 163 168

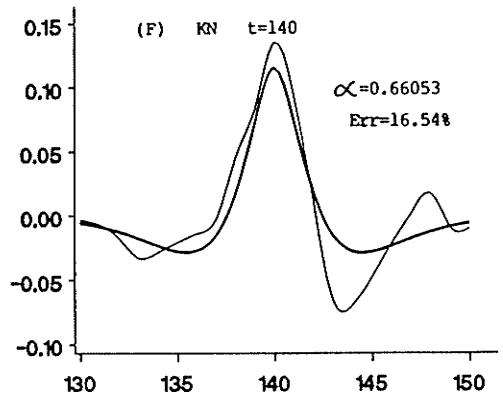

(B) AG t=160
∝ =1.63462
Err=13.78%
0.15
0.10
0.05
0.00
-0.05
150 155 160 165 170

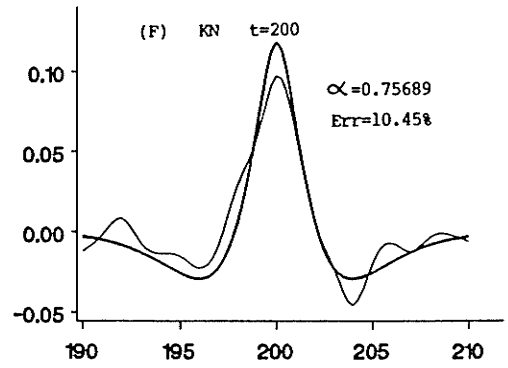

(F) KN t=40
∝=0.72948
Err=28.13%
0.06
0.03
0.00
-0.03
30
35
40
45
50

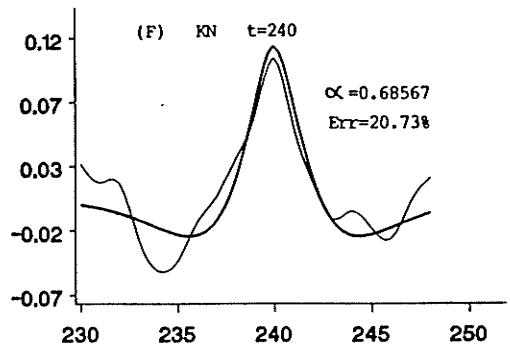

(F) KN t=60
∝=0.68884
Err=20.50%
0.10
0.05
0.00
-0.05
50
55
60
65
70

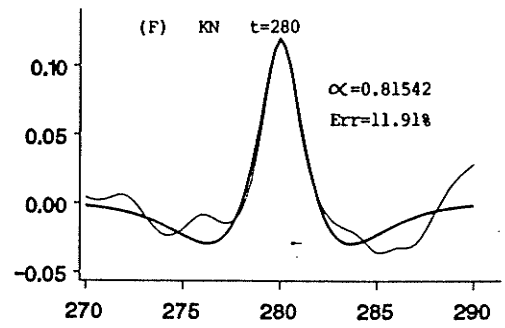

(F) KN t=80
∝=0.66467
Err=18.60%
0.12
0.07
0.03
-0.02
-0.07
70
75
80
85
90

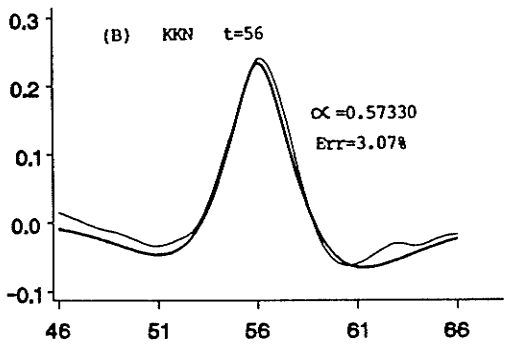

(F) KN t=100
∝=0.67281
Err=16.48%
0.10
0.05
0.00
-0.05
90
95
100
105
110

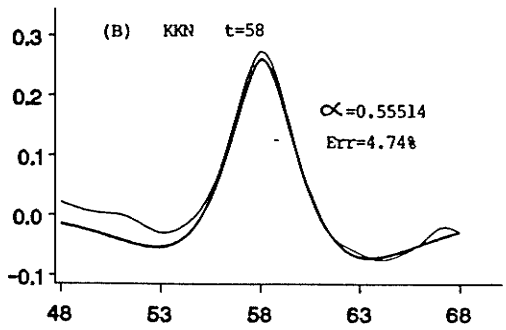

(F) KN t=140
∝=0.66053
Err=16.54%
0.15
0.10
0.05
0.00
-0.05
-0.10
130
135
140
145
150

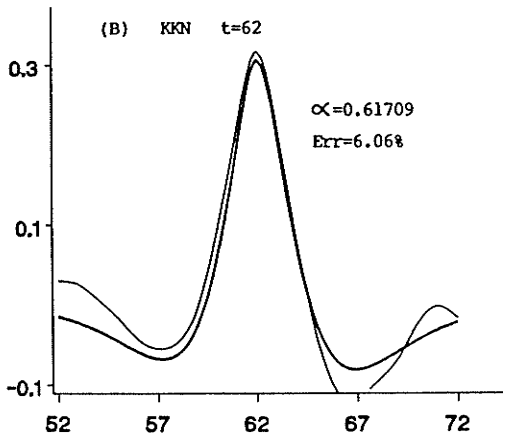

(F) KN t=200
∝=0.75689
Err=10.45%
0.10
0.05
0.00
-0.05
190
195
200
205
210

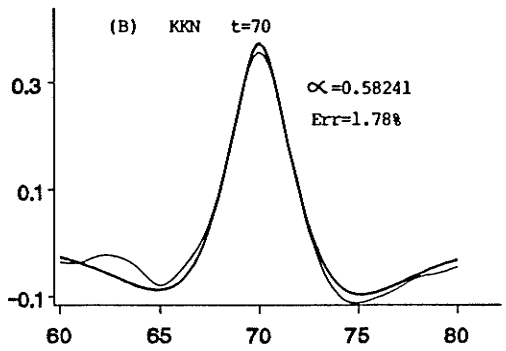

(F) KN t=240
∝=0.68567
Err=20.73%
0.12
0.07
0.03
-0.02
-0.07
230
235
240
245
250

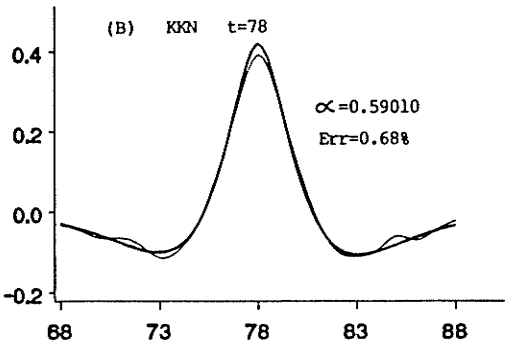

(F) KN t=280
∝=0.81542
Err=11.91%
0.10
0.05
0.00
-0.05
270
275
280
285
290

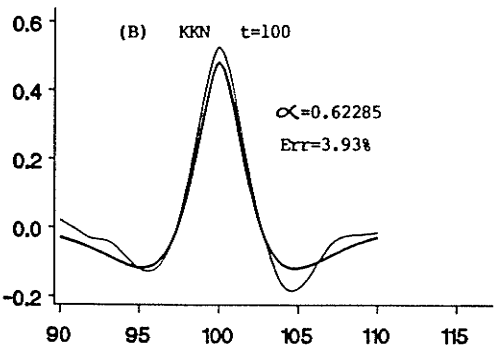

(B) KKN t=56
∝=0.57330
Err=3.07%
0.3
0.2
0.1
0.0
-0.1
46
51
56
61
66

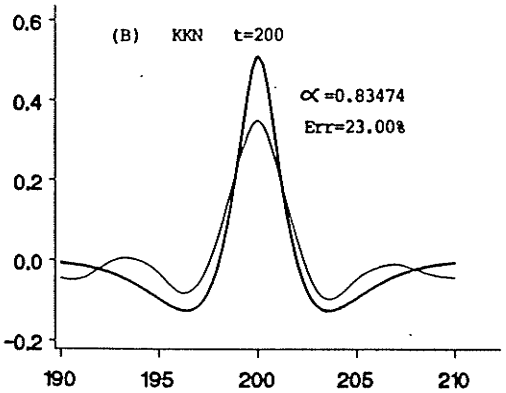

(B) KKN t=58
∝=0.55514
Err=4.74%
0.3
0.2
0.1
0.0
-0.1
48
53
58
63
68

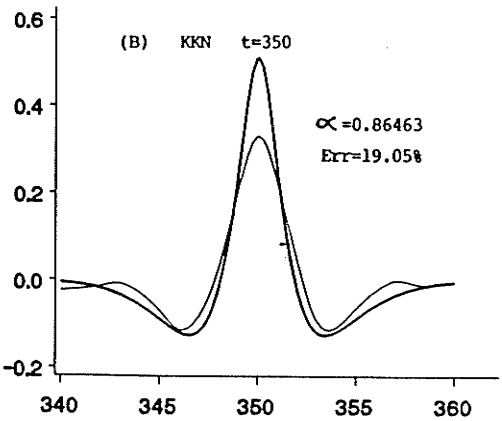

(B) KKN t=62
∝=0.61709
Err=6.06%
0.3
0.1
-0.1
52
57
62
67
72

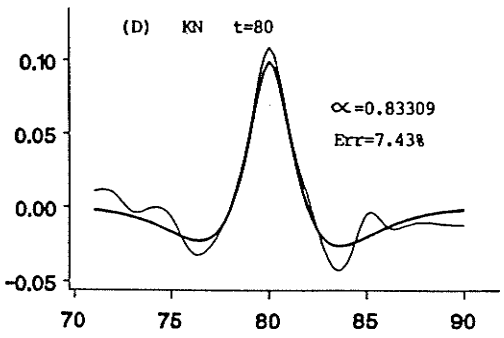

(B) KKN t=70
∝=0.58241
Err=1.78%
0.3
0.1
-0.1
60
65
70
75
80

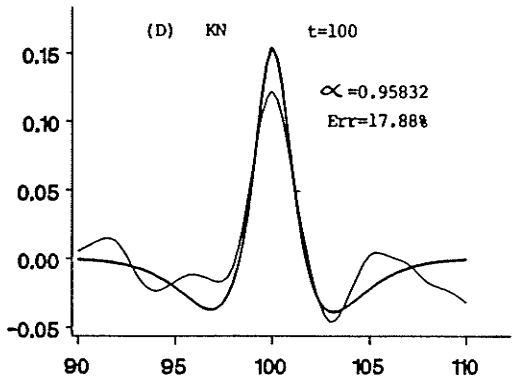

(B) KKN t=78
∝=0.59010
Err=0.68%
0.4
0.2
0.0
-0.2
68
73
78
83
88

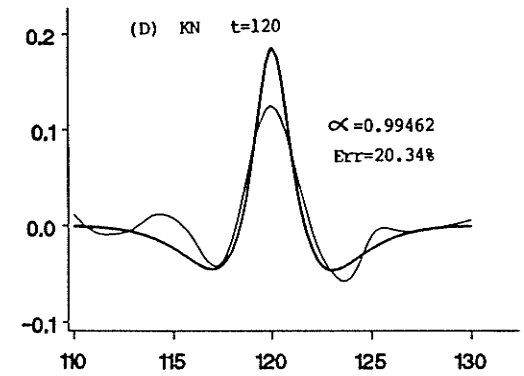

(B) KKN t=100
∝=0.62285
Err=3.93%
0.6
0.4
0.2
0.0
-0.2
90
95
100
105
110
115

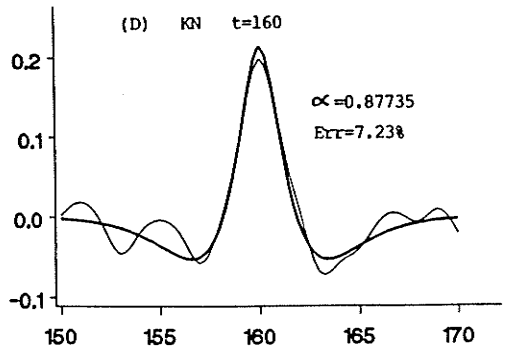

(B) KKN t=200
∝=0.83474
Err=23.00%
0.6
0.4
0.2
0.0
-0.2
190
195
200
205
210

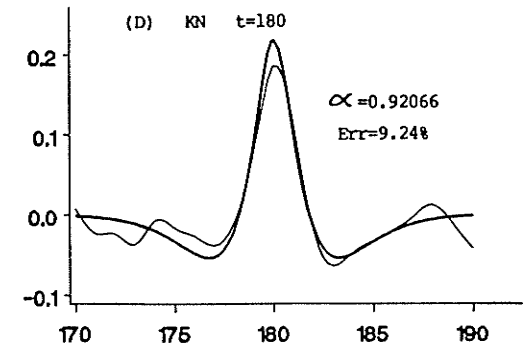

(B) KKN t=350
∝=0.86463
Err=19.05%
0.6
0.4
0.2
0.0
-0.2
340
345
350
355
360

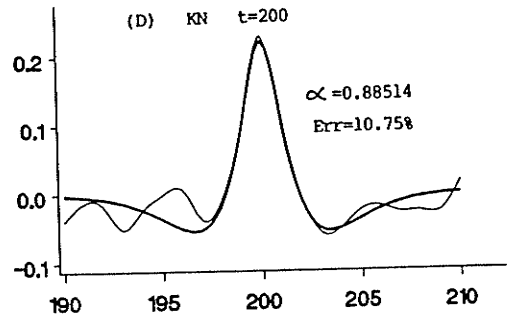

(D) KN t=80
∝=0.83309
Err=7.43%
0.10
0.05
0.00
-0.05
70
75
80
85
90

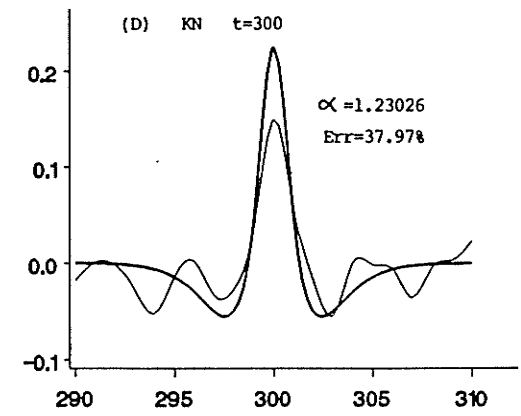

(D) KN t=100
∝=0.95832
Err=17.88%
0.15
0.10
0.05
0.00
-0.05
90
95
100
105
110

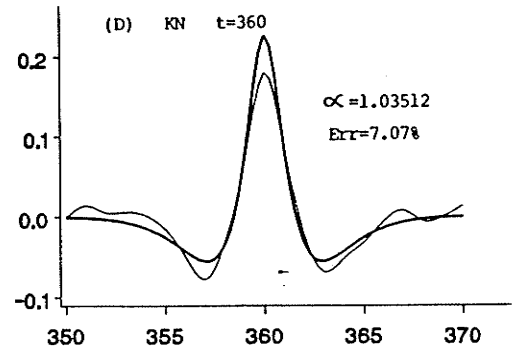

(D) KN t=120
∝=0.99462
Err=20.34%
0.2
0.1
0.0
-0.1
110
115
120
125
130


(D) KN t=160
∝=0.87735
Err=7.23%
0.2
0.1
0.0
-0.1
150
155
160
165
170

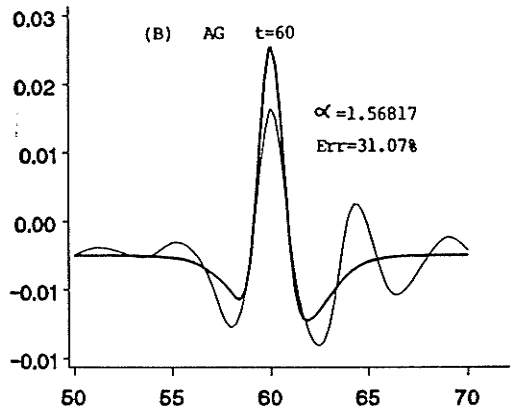

(D) KN t=180
∝=0.92066
Err=9.24%
0.2
0.1
0.0
-0.1
170
175
180
185
190

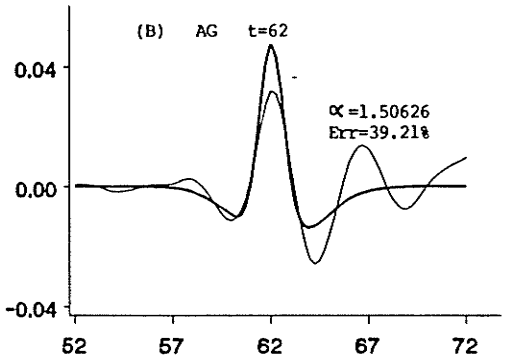

(D) KN t=200
∝=0.88514
Err=10.75%
0.2
0.1
0.0
-0.1
190
195
200
205
210

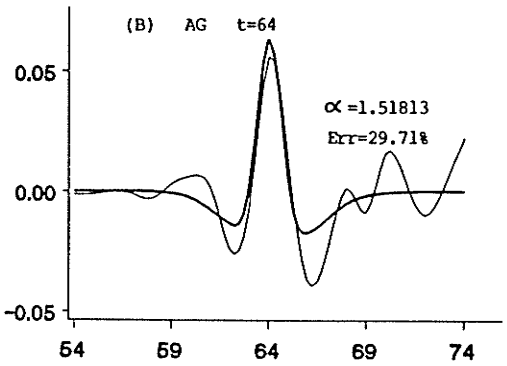

(D) KN t=300
∝=1.23026
Err=37.97%
0.2
0.1
0.0
-0.1
290
295
300
305
310

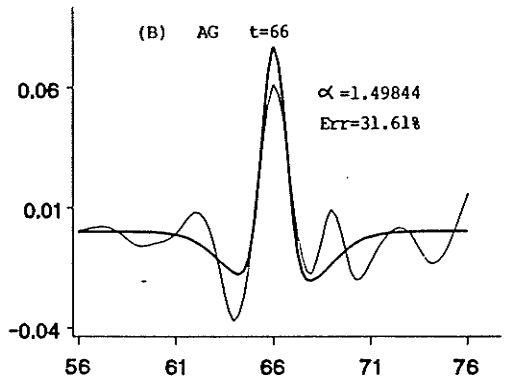

(D) KN t=360
∝=1.03512
Err=7.07%
0.2
0.1
0.0
-0.1
350
355
360
365
370

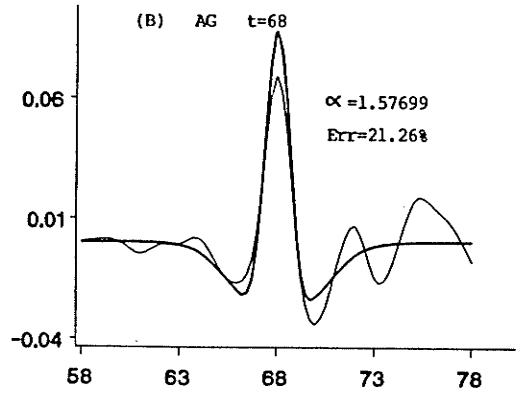
C.4.2 Combined with δe_2

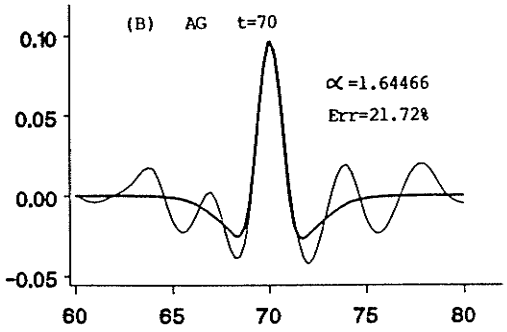


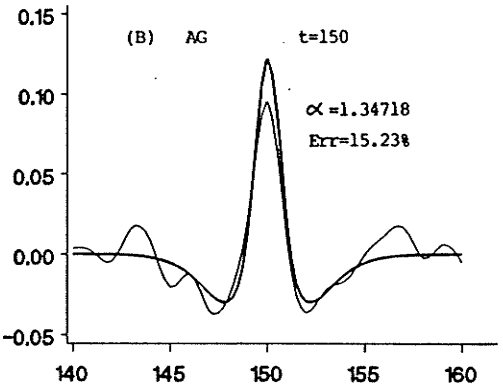


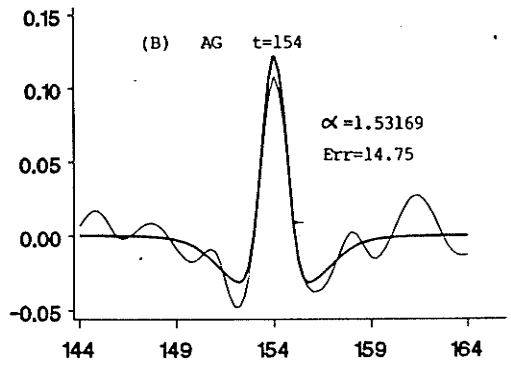


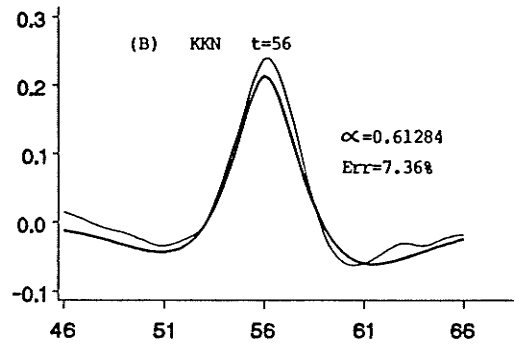


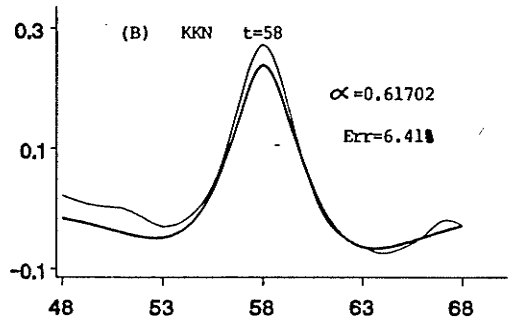


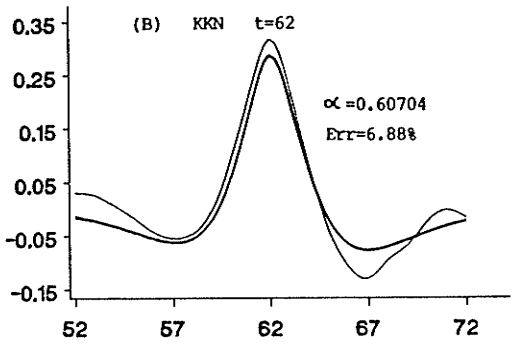


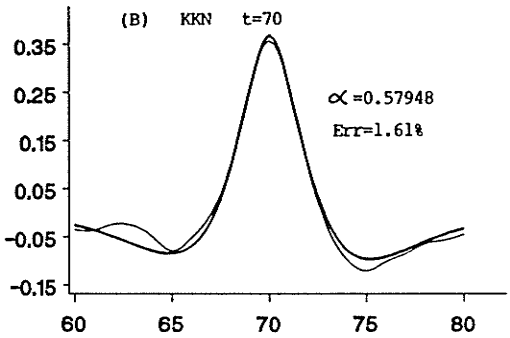


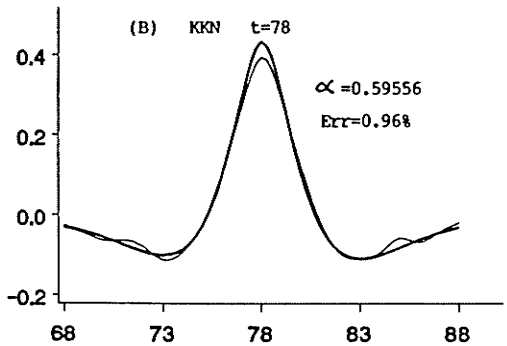


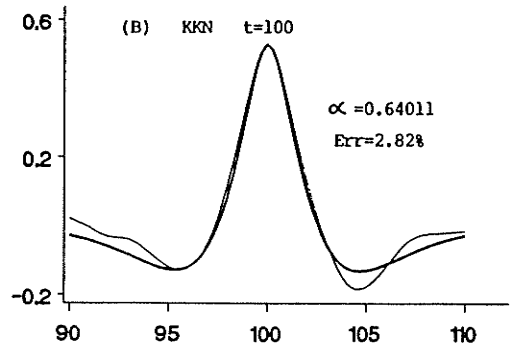

(B) KKN t=56
∝=0.61284
Err=7.36%
0.3
0.2
0.1
0.0
-0.1
46
51
56
61
66

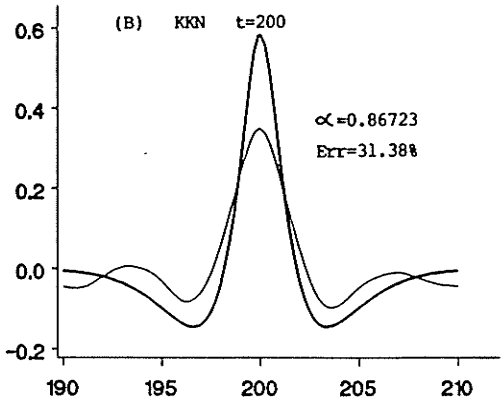

(B) KKN t=58
∝=0.61702
Err=6.41%
0.3
0.1
-0.1
48
53
58
63
68

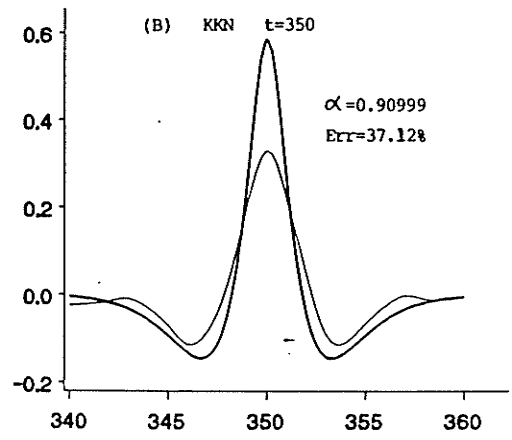

(B) KKN t=62
∝=0.60704
Err=6.88%
0.35
0.25
0.15
0.05
-0.05
-0.15
52
57
62
67
72

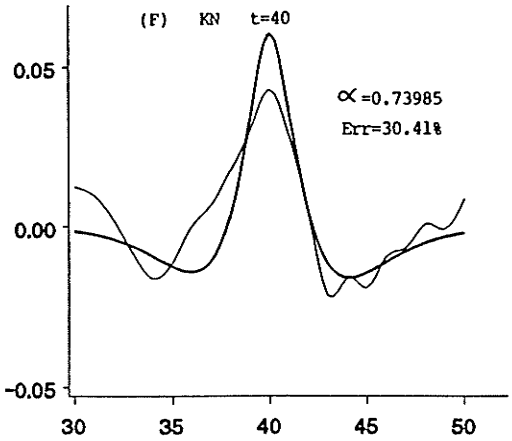

(B) KKN t=70
∝=0.57948
Err=1.61%
0.35
0.25
0.15
0.05
-0.05
-0.15
60
65
70
75
80

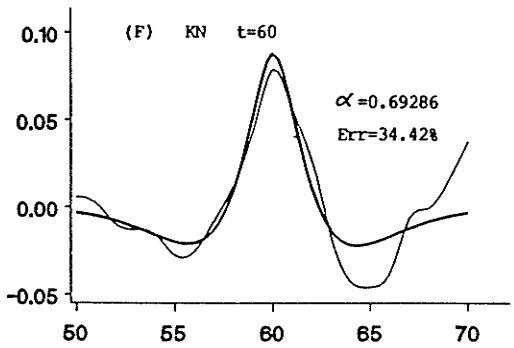

(B) KKN t=78
∝=0.59556
Err=0.96%
0.4
0.2
0.0
-0.2
68
73
78
83
88

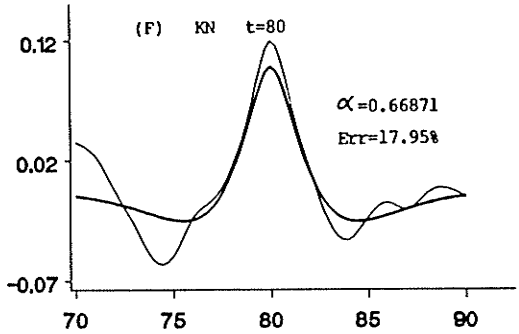

(B) KKN t=100
∝=0.64011
Err=2.82%
0.6
0.2
-0.2
90
95
100
105
110

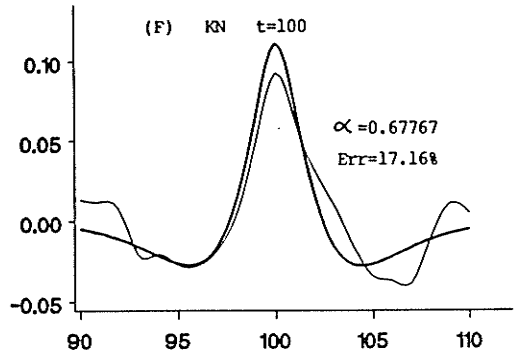

(B) KKN t=200
∝=0.86723
Err=31.38%
0.6
0.4
0.2
0.0
-0.2
190
195
200
205
210

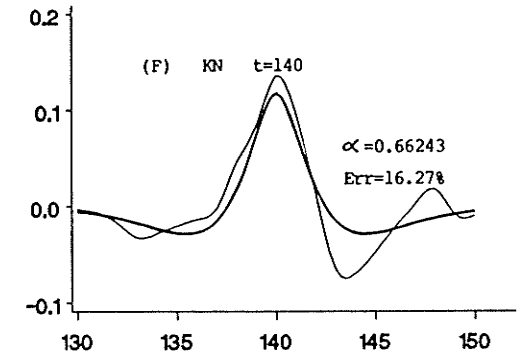

(B) KKN t=350
∝=0.90999
Err=37.12%
0.6
0.4
0.2
0.0
-0.2
340
345
350
355
360

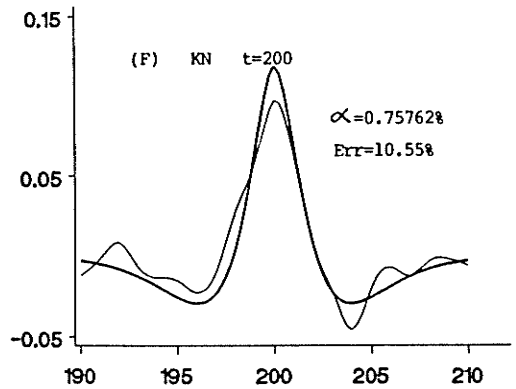

(F) KN t=40
α=0.73985
Err=30.41%
0.05
0.00
-0.05
30
35
40
45
50

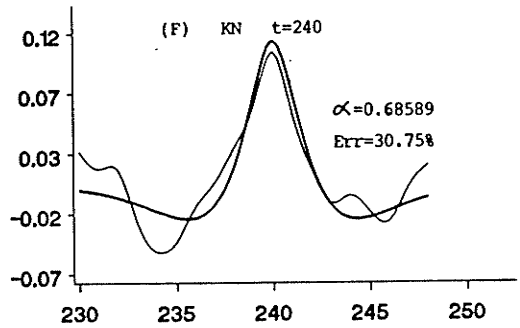

(F) KN t=60
α=0.69286
Err=34.42%
0.10
0.05
0.00
-0.05
50
55
60
65
70

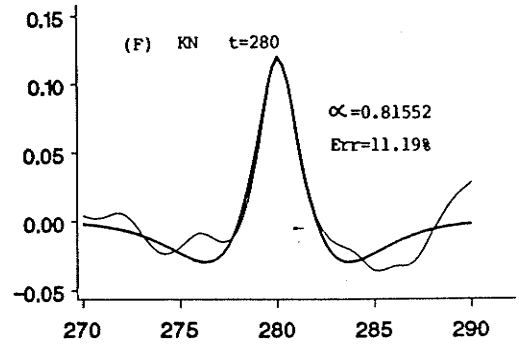

(F) KN t=80
α=0.66871
Err=17.95%
0.12
0.02
-0.07
70
75
80
85
90

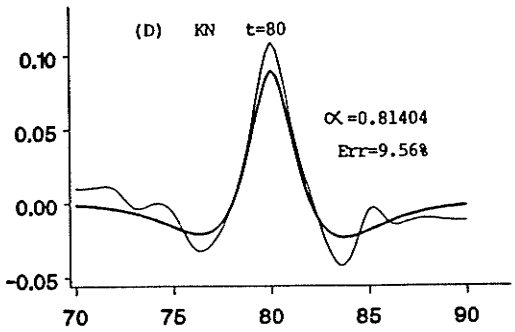

(F) KN t=100
α=0.67767
Err=17.16%
0.10
0.05
0.00
-0.05
90
95
100
105
110

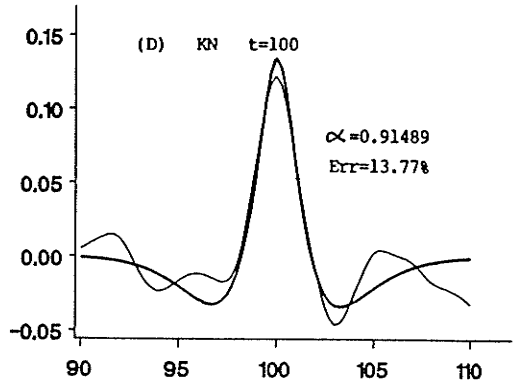

(F) KN t=140
α=0.66243
Err=16.27%
0.2
0.1
0.0
-0.1
130
135
140
145
150

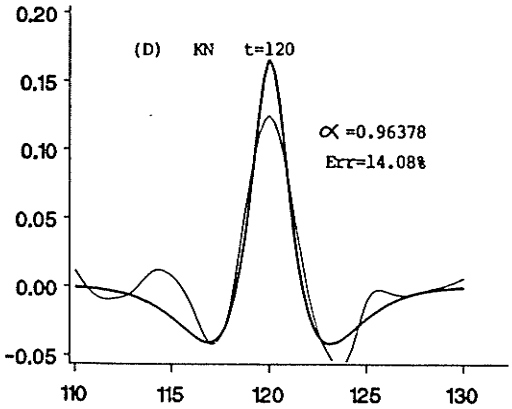

(F) KN t=200
α=0.75762%
Err=10.55%
0.15
0.05
-0.05
190
195
200
205
210

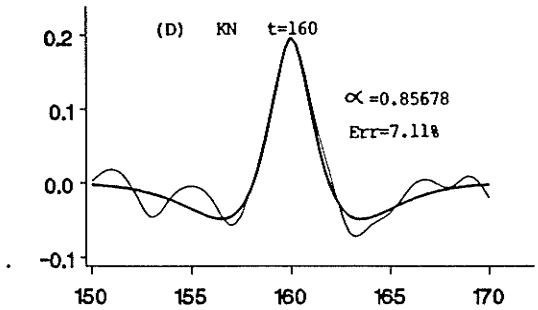

(F) KN t=240
α=0.68589
Err=30.75%
0.12
0.07
0.03
-0.02
-0.07
230
235
240
245
250

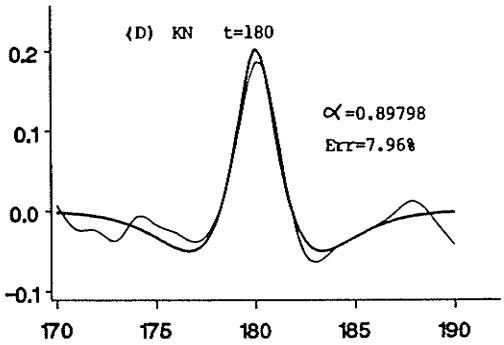

(F) KN t=280
α=0.81552
Err=11.19%
0.15
0.10
0.05
0.00
-0.05
270
275
280
285
290

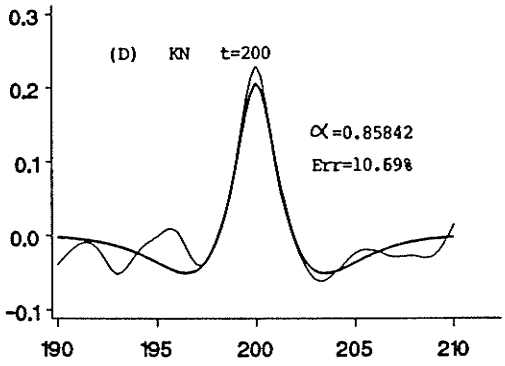

(D) KN t=80
α=0.81404
Err=9.56%
0.10
0.05
0.00
-0.05
70
75
80
85
90

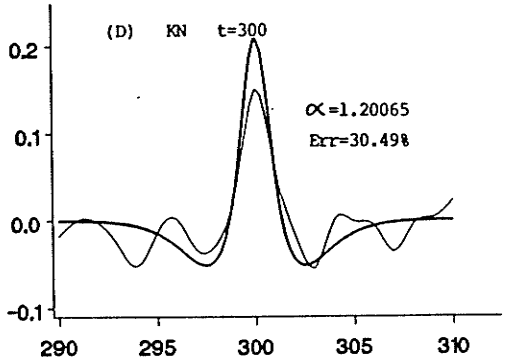

(D) KN t=100
α=0.91489
Err=13.77%
0.15
0.10
0.05
0.00
-0.05
90
95
100
105
110

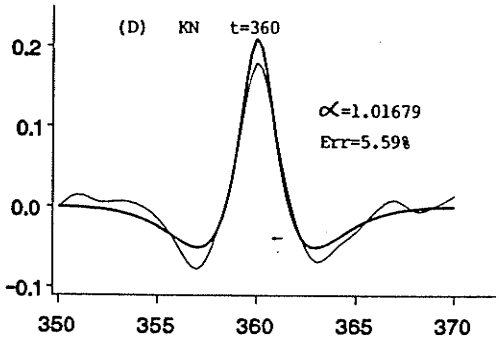

(D) KN t=120
α=0.96378
Err=14.08%
0.20
0.15
0.10
0.05
0.00
-0.05
110
115
120
125
130


(D) KN t=160
α=0.85678
Err=7.11%
0.2
0.1
0.0
-0.1
150
155
160
165
170


(D) KN t=180
α=0.89798
Err=7.96%
0.2
0.1
0.0
-0.1
170
175
180
185
190


(D) KN t=200
α=0.85842
Err=10.69%
0.3
0.2
0.1
0.0
-0.1
190
195
200
205
210


(D) KN t=300
α=1.20065
Err=30.49%
0.2
0.1
0.0
-0.1
290
295
300
305
310


(D) KN t=360
α=1.01679
Err=5.59%
0.2
0.1
0.0
-0.1
350
355
360
365
370

REFERENCE

1. Julius S.& Bendat, Allan G. Piersol, "Random Data: Analysis and Measurement Procedures", Wiley-Interscience, a division of John Wiley & Sons, Inc., N.Y., 1971, p.p.113-114.

2. G. Brody, R.N. Scott, "A Model for Myoelectric Signal Generation", Medical and Biological Engineering, Jan 1974, p.p.29-41.

3. Geoge R. Cooper, Clare D. McGillen, "Probabilistic Methods of Signal and System Analysis", Holt, Rinehart and Wilson, Inc., N.Y., 1971, p.p.115-116.

4. James E. Crouch, "Functional Human Anatomy", 2nd edition, Lea & Febiger, Philadelphia, 1972.

5. Carlo J. DeLuca, "A Model for a Motor Unit Train Recorded During Constant Force Isometric Contraction", Biol. Cybernetics, vol.19, 1975, p.p.159-167.

6. Carlo J. DeLuca, "Physiology and Mathematics of Myoelectric Signals", IEEE Transactions on Biomedical Engineering, vol.26, No.6, June 1979, p.p.

7. John G. Kreifeldt, Sumner Yao, "A Signal to Noise Investigation of Nonlinear Electromyographic Processors", IEEE Transactions on Biomedical Engineering, Vol.BME-21, No.4, July 1974, p.p.298-308.

8. Philip A. Parker, John A. Stuller, R.N. Scott, "Signal Processing for the Multistate Myoelectric", Proceeding of the IEEE, Vol.65, No.5, May 1977, p.p.662-674.

9. George N. Saridis, Thomas P. Gootee, "EMG Pattern Analysis and Classification for a Prosthetic Arm", IEEE Transactions on Biomedical Engineering, Vol.BME-29, No.6, June 1982, p.p.403-412.

10. R.N. Scott, "Technical Note Myoelectric Energy Spectra", Fed & Biol. Engineering, Pergamon Press. 1967, Great Britain, vol.5, p.p.303-305.

11. E. Shwedyk, "Estimation of a Muscle's Force from Its Myo-electric Signal During Quasi-Isometric Contraction", Ph.D thesis, Dept. of Electrical Engineering, University of New Brunswick, Fredericton, N.B., 1973.

12. E. Shwedyk, "EMG Signal", classroom notes.

13. Henry Van Trees, "Detection, estimation, and modulation Theory", John Wiley & Sons, Inc., New York, 1968, p.p.481-488.

14. F.Q. Xiong, E. Shwedyk, "Processing of Nonstationary Myo-electric Signals", Proceedings of the 7th Annual Conference of the IEEE Engineering in Medicine and Biology Society, vol.1, 1985, p.p.279-282.